ONTWIKKELING VAN 'N OUTOMATIESE-TOETSSTELSEL VIR DIE EVALUERING VAN NIE-PROGRAMMEERBARE DIGITALE KRINGE

DEUR

JOSEF JOHANNES SMIT

Skripsie voorgelê ter gedeeltelike voldoening aan die vereistes vir:

Meestersdiploma in Tegnologie

Elektriese Ingenieurswese (Swakstroom)

In die:

Fakulteit Ingenieurswese

aan die:

Technikon OVS

Datum van inlewering: Desember 1992

Studieleier: Mnr. G.D. Jordaan

BEDANKINGS

Graag wil ek die volgende persone en instansies bedank vir hulle bydra tot die suksesvolle voltooiing van die projek:

Aan my Skepper vir die talente wat Hy aan my geskenk het.

My studieleier, Mnr G.D. Jordaan, vir sy hulp en ondersteuning gedurende die projek.

Die Technikon OVS vir die geleentheid aan my gebied om die projek te hanteer.

Aan my ouers, Boet en Mara, vir al die geleenthede wat hulle my gebied het.

Laaste, maar nie die minste nie, my verloofde Adél, wie se aansporing en ondersteuning goud werd was.

Toetsing van elektroniese kringe word deur middel van twee afsonderlike stappe beskryf, naamlik die identifisering van 'n fout en die isolering van die fout. Toetsing behels die toepassing van toetsstimuli (toetsvektore) op die eenheid onder toets (EOT), die meting van die uitset van die EOT en die evaluering van die resultate wat verkry word.

Die vasgeklem-by-waarde foutmodel is gebruik vir die opstelling van 'n foutlys. Hierdie model maak voorsiening dat elke node in die kring vir vasgeklem-by-0 en vasgeklem-by-1 getoets word. Toetsvektore is bepaal vir elke fout wat in die foutlys voorkom.

Die belangrikste aspek van toetspatroongenerering is om te verseker dat 'n fout wat by die inset van 'n komponent voorkom, deur die komponent verplaas word, sodat dit tydens toetsing 'n verandering by die uitset van die komponent teweegbring. Daar bestaan twee stappe indien enige node getoets word. Eerstens moet die insetwaarde so gekies word dat dit die betrokke node aktiveer (beheerbaarheid) en ten tweede moet die effek hiervan na die uitset van die EOT verplaas word (waarneembaarheid).

'n Proses wat as foutineenstorting bekend staan is gebruik om die aantal moontlike foute in die foutlys van sommige kringe te verminder. Foutineenstorting bestaan uit die identifisering van ononderskeidbare- en dominantefoute.

Die evaluering van 'n bord eindig nie by die bepaling van die betrokke toetsvektore nie. Informasie moet saam met die toetsstel ingesluit word om die foutiewe node of komponent in die kring te bepaal aan die hand van die resultate wat deur die toetse verkry is. Foutdiagnosering word gedoen deur middel van 'n besluitnemingsproses aan die hand van die uitslag van die toetse.

Ontwerp-om-te-kan-toets tegnieke - vanaf eenvoudige Ad hoc tegnieke tot strukturele tegnieke en ingeboude-selftoetsing - kan gebruik word om die ontwikkeling van 'n geskikte toetsstel te vereenvoudig.

Die outomatiese-toetsstelsel (OTS) wat ontwikkel is bestaan uit hardeware en sagteware.

Die hardeware van die OTS bestaan uit 'n rekenaar, koppelvlakkaart, en die hegstuk tussen die koppelvlakkaart en die EOT. Vir die gebruik as 'n outomatiese-toetsstelsel kan van enige IBM PC/XT/AT of IBM aanpasbare persoonlike-rekenaar gebruik gemaak word. 'n Koppeling tussen die rekenaar en die EOT moet bestaan vir die oordrag van die toetsvektore vanaf die rekenaar na die EOT en vir die versameling van die resultate vanaf die EOT. Vir hierdie doel word van die PC-14A koppelvlakkaart gebruik gemaak. Die OTS maak voorsiening om borde wat van 'n randkonnekteerder voorsien is te kan toets.

Die sagteware van die OTS bestaan uit die toetsprogram self en 'n databasis (toetsstel) vir elke bord wat deur die OTS getoets kan word. Die toetsprogram word gebruik vir die

opstelling van 'n toetsstel asook die uitvoering van die toetse.

Ter evaluering van die OTS is twee kringe gebou en 'n toetsstel vir elk bepaal. Ter evaluering van die effektiwiteit van die toetsstelle is al die foute, soos in die betrokke foutlyste uiteengesit, op die kringe gesimuleer. Alle foute wat gesimuleer is, is wel deur die OTS geïdentifiseer nadat die betrokke toetsstel op die bord toegepas is.

Aangesien alle kringnodes voorkom in die betrokke foutlyste - en alle foute in die foutlyste wel geïdentifiseer is - kan gesê word dat 100% foutdekking in albei gevalle behaal is.

SINOPSIS

Testing of electronic circuits can be divided into two steps, namely the identification of a fault and the isolation of the fault. Testing comprises the application of test stimuli (test vectors) to the unit under test (UUT), the measurement of the output of the UUT and the evaluation of the obtained results.

The stuck-at-value fault model was used to determine the fault list. This model makes provision for the testing of each node in the circuit stuck-at-0 and stuck-at-1. Test vectors were determined for each fault in the fault list.

The most important aspect of test pattern generation is to ensure that a fault appearing at the input of a component produces an effect at the output of the component during testing. There are two steps in the testing of any node. Firstly the input must be determined so as to activate the node being tested (controllability) and secondly the effect of this must be transferred to the output of the circuit (observability).

A process named fault collapsing was used to reduce the size of the fault list of a few circuits. Fault collapsing consists of identifying indistinguishable and dominant faults.

The evaluation of a circuit does not end with the determination of the test vectors. Fault information has to

be included in the test set to determine the faulty node or component in the circuit. Fault diagnosis is done by means of a decision making process according to the test results obtained.

Design for testability techniques (Ad hoc techniques, structural techniques and built-in self testing) have been investigated as means to simplify the development of a test set.

The automatic test system (ATS) developed consists of hardware and software.

The hardware of the ATS consists of a computer, interface and a fixture between the interface and the UUT. Any IBM PC/XT/AT or IBM compatible PC can be used for the ATS. A PC-14A interface card is used as interface between the computer and the UUT for the transfer of test vectors from the computer to the UUT and the collecting of results from the UUT. Provision has been made for the testing of cards equipped with an edge connector.

The ATS software consist of the test program and a data base for each circuit to be tested with the ATS. The test program is used for the composition of the test set as well as the execution of the tests.

To evaluate the ATS two circuits have been built and a test set determined for each. All the faults, as given in the relevant fault lists, were simulated on the circuits and were

identified during application of the test sets to the circuits.

Since all nodes were included in the fault lists, and since all faults were identified, it can be said that a fault coverage of 100% has been obtained in both cases.

INHOUDSOPGAWE

BLADSY

LYS VAN FIGURE

BLADSY

HOOFSTUK 2

HOOFSTUK 3

LYS VAN TABELLE

BLADSY

Inleiding

1.1 Probleemstelling

'n Mate van evaluering van die kwaliteit of werking van 'n vervaardigde artikel vind altyd plaas voor verspreiding van sodanige artikel. Dit word gedoen om te verseker dat die artikel korrek funksioneer. In die verlede het toetsing eenvoudig die visuele inspeksie van die finale produk behels.

Met die vervaardiging van elektroniese stelsels het toetsing toenemend gevorderd geraak aangesien visuele inspeksies nie meer voldoende was vir die evaluering van die produk nie. Nog steeds was dit egter nie so 'n groot probleem nie aangesien 'n produk, soos byvoorbeeld 'n radio, as korrek aanvaar kan word indien na een stasie op elke band geluister word en hierdie geselekteerde stasies korrek funksioneer.

Sedert die ontwikkeling van mikroëlektronika het toetsing meer gevorderd geword, veral by stelsels wat gebruik maak van dataprosessering. Hierdie stelsels verwerk data wat by die inset van die stelsel gelewer word volgens 'n bepaalde funksie om 'n bepaalde uitset te lewer. Die stelsels is gewoonlik baie groot en toetsing word bemoeilik as gevolg van drie redes:

i) Vir meganiese redes word 'n stelsel gewoonlik opgedeel in kleiner sub-eenhede en dié sub-eenhede het gewoonlik geen losstaande funksie nie.

ii) Die stelsel is gewoonlik van so 'n aard dat die inset en uitset slegs uit 0e en 1e bestaan. Deur slegs na 'n enkele uitset van 0e en 1e te kyk onder 'n enkele insettoestand kan nie gesê word of die stelsel normaal funksioneer al dan nie.

iii) Aangesien die uitset afhanklik is van die inset bestaan daar 'n behoefte om spesifieke toetsdata op die kring uit te oefen. Toetsdata moet so saamgestel word dat die werking van die kring onder alle toestande gesimuleer word.

1.2 Doelwit van die ondersoek

In die lig van bogenoemde probleme, en om terselfdertyd die koste van die vervaardigingsproses so laag moontlik te hou, is outomatiese toetsing dus wenslik om kringe te kan toets. Outomatiese toetsing moet in die vervaardigingsproses gebruik word aangesien die vervaardiger nie die tyd om die toetse met die hand uit te voer, of die opgeleide personeel om die toetse uit te voer, kan bekostig nie.

'n Poging is aangewend om 'n outomatiese-toetsstelsel te ontwikkel om die werking van nie-programmeerbare digitale kringe te verifieer. Indien enige afwyking van die verlangde werking geïdentifiseer word, word 'n aanduiding gegee van alle moontlike foutiewe komponente in die kring.

1.3 Hipotese

'n Outomatiese-toetsstelsel (OTS) kan ontwikkel word om die werking van nie-programmeerbare digitale kringe vas te stel, en om foutiewe komponente te identifiseer indien die werking van die kring nie na wense is nie.

1.4 Afbakening van die studieterrein

Soos die titel aandui is hierdie studie toegespits op nie-programmeerbare digitale kringe. Die ontwikkeling van toets-vektore vir analoog en digitale komponente verskil drasties en daarom is besluit om hierdie studie slegs op digitale komponente toe te spits.

Komponente wat in bogenoemde kategorie val - en wat gebruik is om 'n aantal kringe te bou vir toetsing - is onder andere EN-, OF-, NEN-, NOF- en eksklusiewe OF-hekke, omkeerders, multipleksers en 'n verskeidenheid wipkringe.

1.5 Metode van navorsing

Die projek kan in 'n aantal diskrete stappe verdeel word. 'n Kort beskrywing van elke stap word gegee.

1.5.1 Rekenaar en koppelvlakkaart

Vir die gebruik as outomatiese toetsstelsel kan van enige IBM PC/XT/AT of IBM aanpasbare persoonlike-rekenaar gebruik gemaak word. Vir die oordraging van

toetsvektore vanaf die rekenaar na die eenheid onder toets moet 'n koppelvlak verskaf word. Aangesien binêre data-oordraging benodig word, word van die PC-14A koppelvlakkaart gebruik gemaak. 'n Randkonnekteerder dien as fisiese verbinding tussen die PC-14A en die EOT.

1.5.2 Algoritmes en sagteware

Die sagteware van die OTS bestaan uit twee dele, naamlik die toetsprogram self en 'n databasis (toetsstel) vir elke bord wat deur die stelsel getoets kan word. Die sagteware vir die toetsprogram is in Turbo Pascal geskryf.

Om alle foute wat in 'n kring kan voorkom te identifiseer moet die korrekte toetspatroon op die kring uitgevoer word. Vir die bepaling van die toetspatroon (toetsvektore) is 'n dieptestudie uitgevoer oor die tegnieke van toetspatroongenerering.

1.5.3 Bou en toetsing van toetskringe

Twee toetskringe is gebou om die werking van die stelsel asook die effektiwiteit van die toetsstelle te toets. Aangesien die kringe net vir die evaluering van die stelsel gebou is, is besluit om van eksperimenteringsbord gebruik te maak.

Nadat die toetsvektore saamgestel, en die kringe gebou is, is foute op die kringe gesimuleer om vas te stel wat die efektiwiteit van die toetsstelsel is.

1.6 Probleme ondervind

Die grootste probleem is ondervind by die bepaling van 'n toetsstel vir die volgordekring. Dié probleem ontstaan aangesien die uitset van die kring nie net bepaal word deur die huidige insette nie, maar ook deur die stand waarin die kring verkeer as gevolg van die invloed wat vorige insette op die kring gehad het.

'n Probleem wat ondervind is by die skryf van die sagteware is om te verseker dat volledige fouttoetsing uitgevoer word op alle data wat deur die gebruiker aan die stelsel verskaf word. Indien enige data verskaf aan die stelsel buite die normale grense vir die spesifieke data of besondere kring val, moet dit deur die stelsel geïdentifiseer word en die nodige stappe diensooreenkomstig geneem word.

TOETSTEGNIEKE IN DIGITALE KRINGE

2.1 Wat behels elektroniese toetsing?

Toetsing van elektroniese kringe kan deur middel van twee afsonderlike stappe beskryf word, naamlik die identifisering van 'n fout, en die isolering van die fout. By die identifisering van 'n fout word daar vasgestel of 'n kring, stelsel of substelsel foutief is. Met die isolering van 'n fout word daar gepoog om die kleinste moontlike foutiewe eenheid, naamlik 'n komponent, te bepaal (Pynn, 1986:45).

Vir toetsing word drie elemente benodig naamlik die eenheid onder toets (EOT), die toetsstimuli en die bekende foutvrye uitset van die EOT. Toetsing behels toepassing van toetsstimuli op die EOT, meting van die uitset van die EOT en evaluering van die resultate wat verkry word.

2.2 Parametriese- en logikatoetsing

By parametriesetoetsing bestaan die insetstimuli uit kombinasies van byvoorbeeld spanning, stroom en tyd, en die uitset die meting van analoog parameters soos bandwydte, versterking en frekwensieweergawe.

Logikatoetsing daarteenoor behels toepassing van logikaseine by die inset van die EOT en evaluering van die binêre waardes wat by die uitset verkry word. Die binêre waardes neem

byvoorbeeld die logikavlakke van 0V en 5V aan. 'n Baie algemene verskynsel is dat 'n kring 'n kombinasie van parametriese- en logikatoetsing benodig om die kring ten volle te kan toets.

Voortaan sal slegs na logikatoetsing verwys word.

2.3 Metodes van toetsing

Een van twee metodes word gewoonlik gebruik met die toetsing van 'n kring. Die metode wat gebruik word sal afhang van die soort kring wat getoets moet word asook die toerusting of meetinstrumente wat beskikbaar is. Toetsing kan op een van die volgende metodes gedoen word:

2.3.1 Toetsing met die hand

Wanneer 'n kring met die hand getoets word, word daar hoofsaaklik van instrumente soos ossilloskope, logika toetspenne, logika pulsers en logika analiseerders gebruik gemaak. Hier word hoofsaaklik van die beginsel van seinsporing gebruik gemaak waar die sein deur die kring gevolg word nadat dit by die inset van die kring aangelê is.

Hierdie metode van toetsing het egter baie nadele. Die twee grootstes hiervan is dat baie tyd in beslag geneem word om die kring te toets, en dat daar van goed opgeleide personeel gebruik gemaak moet word om die toetsing te behartig. Boonop kan alle logikastate nie

met die hand uitgevoer word nie en dus is dit moontlik dat daar foute kan deurglip wat nie as sulks geïdentifiseer word nie.

2.3.2 Outomatiese toetsing

As gevolg van die geweldige kompleksiteit van elektroniese kringe het toetsing met die hand in baie gevalle onmoontlik geword en word outomatiese toetsing gebruik. Aangesien outomatiese toetsing 'n proses behels wat baie herhaling vereis, word rekenaars gebruik om die toetsing te verrig.

In outomatiese-toetsstelsels (OTS) staan die rekenaar bekend as die beheerder aangesien dit gebruik word vir die berging, beheer en uitvoer van die toetsprosedure (Healy, 1981:25).

'n OTS bestaan hoofsaaklik uit drie komponente, naamlik 'n rekenaar, 'n koppelvlak tussen die rekenaar en die EOT en sagteware. Die sagteware bestaan verder uit twee elemente, naamlik die toetsprogram self en 'n addisionele databasis vir elke bord wat deur die stelsel getoets kan word. Die databasis bestaan basies uit die toetsvektore, die foutvrye uitset van die EOT en foutinligting.

2.4.1 Funksionele toetsing

Met funksionele toetsing word aanvaar dat die bord korrek is indien die funksie van die bord korrek uitgevoer word (Randell en Treleaven, 1983:179). Funksionele toetsing kan onderverdeel word in statiese en dinamiese toetsing.

2.4.1.1 Statiese toetsing

Statiese toetsing is gewoonlik goedkoper as dinamiese toetsing en word algemeen gebruik (Capillo, 1990:333). In plaas daarvan om elke komponent afsonderlik te toets word die bord as 'n geheel getoets.

Die insetstimuli, of toetsvektore, word teen 'n laer as normale werkspoed toegepas. Indien die funksie van die bord korrek uitgevoer word, word daar aanvaar dat die bord korrek is. Vir die koppeling tussen die EOT en die OTS word gewoonlik van 'n randkonnekteerder gebruik gemaak. Die randkonnek-teerder is deel van die finale produk en word gebruik vir die verbinding tussen die bord en die stelsel.

Die nadele van statiese toetsing is dat tydafhanklike foute straks nie geïdentifiseer sal

word nie, '.. lang uitvoertyd indien 'n groot aantal toetsvektore uitgevoer moet word en dat slegs een fout geïdentifiseer word vir elke uitvoering van die toets.

2.4.1.2 Dinamiese toetsing

Met dinamiese toetsing funksioneer die bord teen normale klokspoed terwyl die uitset gemonitor word. Dié tipe toetsing word hoofsaaklik gebruik vir borde wat komponente soos LSG (RAM), LAG (ROM) en mikro-verwerkers bevat. 'n Bed-van-naalde koppeling word gewoonlik nie gebruik nie as gevolg van hoë kapasitiewe ladings wat kan voorkom en as gevolg van ruis oorwegings (Turino, 1990:305).

Die voordele verbonde aan dinamiese toetsing is dat tydafhanklike foute wel geïdentifiseer sal word en dat die toets baie vinnig uitgevoer word. Die nadele is dat die koste aan toetsapparaat baie hoog is en dat net een fout geïdentifiseer word vir elke uitvoering van die toets.

2.4.2 In-die-kring toetsing

In-die-kring toetsing oorweeg nie die werking van die kring as geheel nie, maar eerder die werking van elke individuele komponent (Hinch, 1988:428). Vir in-die-kring toetsing word van die bed-van-naalde koppeling gebruik gemaak om kontak met al die nodes in die kring

te maak. Toetse word uitgevoer deur die toetsseine oor kombinasies van komponente toe te pas en die resultate te lees om sodoende 'n foutiewe komponent te identifiseer.

2.4.3 Dinamieseverwysing toetsing

Dinamieseverwysing toetsing is 'n tegniek waar die toetsvektore aan die EOT en 'n verwysingsbord gelyktydig gevoer word. Daar word aanvaar dat geen foute op die verwysingsbord bestaan nie. Die uitsette van die twee borde word deur 'n eksklusiewe OF-hek gestuur en so word die EOT gemonitor vir enige foute wat mag voorkom.

Voordele verbonde aan dié soort toetsing is dat groot hoeveelhede toetsvektore teen hoë snelhede uitgevoer kan word, geen LSI modellering benodig word nie en die werking van die bord word volledig geverifieer. Nadele is dat toetsprogramgenerering met die hand gedoen moet word en dat 'n verwysingsbord benodig word vir die toetsing.

2.4.4 Warmnabootsing (Hot mock-up)

Dié metode van toetsing word gewoonlik gebruik om een soort bord - wat deel uitmaak van 'n stelsel van verskeie borde - te toets. 'n Volledige stelsel word opgebou en die bord wat getoets moet word, word in die stelsel geplaas. Al die borde is korrek behalwe die bord onder toetsing, waarvan die toestand onbekend is. Indien

die stelsel as 'n geheel korrek funksioneer word aanvaar dat die bord korrek is. Dié tegniek is baie eenvoudig en is ook baie goedkoop aangesien produksieoorskotte gebruik kan word om die stelsel op te bou.

Die grootste nadeel van die tegniek is dat geen of weinig diagnosering van 'n fout vekry word. Boonop word goed opgeleide personeel benodig en die toets neem baie tyd in beslag.

2.5 Tegnieke ter bepaling van toetsvektore

Toetstegnieke - soos funksionele toetsing, dinamieseverwysing toetsing en warmnabootsing - het een gemeenskaplike nadeel, naamlik die onvermoë om te bepaal watter komponent in die kring foutief is. Indien foutopsporing gedoen word, moet daar altyd gepoog word om die kleinste moontlike foutiewe eenheid of komponent te bepaal. Metodes moet dus ontwikkel word om 'n digitale kring te toets en terselfdertyd die fout sover moontlik te isoleer.

Toetsing van 'n kring behels die herhaaldelike toepassing van logikawaardes (1e en 0e) by die primêre insette en die waarneming van die resultate by die primêre uitsette van die kring. Die primêre insette en -uitsette van 'n kring is die nodes wat direk aan die randkonnekteerder verbind is.

Om te bepaal of die resultate wat tydens 'n toets verkry word korrek is al dan nie, moet die foutvrye uitset van die kring vir elke insetkondisie bekend wees. Elke toets - wat bestaan

uit 'n insetwaarde tesame met die foutvrye uitsetwaarde - staan as 'n toetspatroon of toetsvektor bekend. 'n Volledige stel toetsvektore, wat benodig word om 'n kring ten volle te kan toets, staan as 'n toetsstel bekend.

2.5.1 Toetspatroongenerering

Die mees tydrowende en uitdagende deel van toetspatroon-generering is die bepaling van die vektore, sodat wanneer dit toegepas word op die EOT, die kring ten volle en deeglik getoets sal word (Hakim, 1989:154). Die toetsstel moet so saamgestel word dat indien 'n bord met 'n defek getoets word, die bord ten minste een van die toetse sal faal.

2.5.1.1 Funksionele toetspatroongenerering

Funksionele toetspatroongenerering behels die ontwikkeling van 'n toetsstel volgens die funksie van die kring, sonder om te weet wat die presiese hekvlak implementering van die kring is (Pitchumani en Soman, 1988:756). 'n Voorbeeld van 'n funksionele toets is die van 'n EN-hek met drie insette. Die toetsstel sal uit agt toetsvektore bestaan volgens die waarheidstabel van die hek. Hierdie sal 'n volledige toets van die hek wees aangesien alle moontlike insetkombinasies op die hek uitgeoefen word.

Die gebruik van 'n kring se totale waarheidstabel kom aantreklik voor, maar soos deur Morant (1990:120) en Wilkens (1986:24) aangetoon, is dit onmoontik om die metode toe te pas op 'n kring van redelike grootte. Dus word 'n volledige toets nooit gebruik nie, behalwe by baie klein kringe.

2.5.1.2 Strukturele toetspatroongenerering

Funksionele toetspatroongenerering word gebruik om vas te stel of 'n kring normaal funksioneer, terwyl strukturele toetspatroongenerering gebruik word om te bepaal of die kring foutief is (Wilkens, 1986:25).

'n Fisiese defek in 'n kring veroraak 'n foutiewe uitset en aangesien die toepassing van 'n toetsvektor nie die defek sal waarneem nie, maar eerder die foutiewe uitset, moet die fout geïnterpreteer word as veroorsaak deur 'n sekere defek. Die verhouding tussen 'n defek en 'n fout word as die foutmodel beskryf.

2.5.2 Foutmodel

Bykans alle toetstegnieke het as uitgangspunt 'n foutmodel wat gebaseer word op tipiese foute wat in die praktyk voorkom (Hakim, 1989:91). Met elke tegniek word gepoog om toetse te genereer wat alle foute suksesvol sal dek.

Die sukses van 'n toets is direk afhanklik van die foutmodel wat gekies word. Hoe eenvoudiger die foutmodel - deur byvoorbeeld vir slegs een tipe fout te toets - hoe makliker is dit om 'n toets te genereer en hoe hoër is die waarskynlikheid dat 'n fout sal deurglip sonder om geïdentifiseer te word.

Die volgende is tipiese foutmodelle in algemene gebruik:

2.5.2.1 Vasgeklem-by-waarde model

Die mees algemene model wat gebruik word vir logikafoute is die vasgeklem-by-waarde model (Yarmolik, 1990:1). Met hierdie foutmodel word aangeneem dat alle defekte in 'n kring sal veroorsaak dat 'n node, hetsy inset of uitset, permanent vasgeklem sal word by of 'n een (1) of 'n zero (0). Om vir sodanige fout te toets moet elke node in die kring om die beurt vanaf 'n 1 na 'n 0 en vanaf 'n 0 na 'n 1 geskakel word terwyl die uitset gemonitor word. 'n Geldige aanname is dat indien 'n enkele fout in 'n kring voorkom, slegs een node beïnvloed sal word.

Alhoewel daar foute bestaan wat nie 'n node by 'n waarde sal vasklem nie, word die vasgeklem-by-waarde model universeel in die toetsindustrie gebruik as die basis vir strukturele toetspatroongenerering. Dit dien ook as 'n standaard indien die kwaliteit

van 'n toetsstel bepaal moet word (Wilkens, 1986:26).

2.5.2.2 Bruggingsfoute model

'n Bruggingsfout kom voor in 'n kring indien daar 'n kortsluiting tussen twee of meer bene van 'n komponent - of bane op 'n bord - bestaan. Op 'n gedrukte stroombaan word kortsluitings hoofsaaklik deur soldeerselspatsels veroorsaak. Vir positiewe logika neem die bruggingsfout die funksie van 'n EN-hek aan en vir negatiewe logika die funksie van 'n OF-hek (Kodandapani en Pradhan, 1980:55).

Bruggingsfoute het 'n logika uitwerking op die kring wat verskil van dié van die vasgeklem-by-waarde foute, aangesien die geaffekteerde node by bruggingsfoute steeds albei logika waardes kan aanneem. Indien albei die nodes wat gekortsluit is dieselfde foutvrye waarde (0 of 1) moet aanneem, sal die werking van die kring nie deur die defek beïnvloed word nie. Indien die twee nodes egter verskillende waardes op enige tydstip moet aanneem, sal dit wel die werking van die kring beïnvloed.

Die invloed wat dit op die kring het hang af van die tegnologie wat gebruik word. Dit is bekend dat vir emittergekoppeldelogika (ECL), resistor-transistor logika (RTL) en diode-transistor logika (DTL) die bruggingsfout as 'n bedraad-EN of 'n bedraad-OF

funksie voorgestel kan word (Kodandapani en Pradhan, 1980:55). Vir transistor-transistor logika (TTL) hekke veroorsaak die bruggingsfout 'n EN-hek funksie en dus sal 'n LAAG altyd oorheers.

By komplementêre simmetrie metaaloksied halfgeleier (CMOS) toestelle sal die spanning by die kortsluiting afhanklik wees van die totale effektiewe weerstand wat tussen VDD en grond bestaan. Indien twee nodes onder normale werking teenoorgestelde waardes aanneem, en 'n kortsluiting bestaan daartussen, sal dit 'n hoë toevoerstroom veroorsaak.

Die volgende probleme ontstaan indien bruggingsfoute in 'n kring voorkom:

i) Die voorkoms van 'n bruggingsfout in 'n kring kan die vasgeklem-by-waarde model ongeldig maak.

ii) Die voorkoms van 'n kortsluiting tussen die inset en die uitset van 'n komponent veroorsaak 'n terugvoerlus wat veroorsaak dat 'n gewone kombinasiekring in 'n volgordekring kan verander, of dat die kring kan begin ossilleer.

As gevolg van die groot hoeveelheid kombinasies wat tussen die bane van 'n gedrukte stroombaan voorkom is dit nie moontlik om vir kortsluitings tussen alle kombinasies te toets nie. So het 'n kring met 1000 nodes bykans 'n halfmiljoen kombinasies tussen bane (Wilkens, 1986:38). Derhalwe is dit die gebruik om

nadat die bordutley gedoen is, te besluit tussen watter bane vir bruggingsfoute getoets gaan word.

2.5.2.3 Omgekeerdefoute model

'n Omgekeerdefout het betrekking op 'n fisiese defek in 'n kring wat veroorsaak dat 'n vals omkeerder by die inset of uitset van 'n hek verskyn (Yarmolik, 1990:3). Omgekeerde foute en vasgeklem-by-waarde foute, word gewoonlik gebruik om 'n volledige foutmodel saam te stel vir digitale kringe.

2.5.2.4 Vermengingsfoute (Mix-up faults) model

'n Vermengingsfout kom voor vanweë foutiewe verbindings in 'n digitale kring en word veroorsaak deur die verbreking van ontwerpreëls of vervaardigingsdefekte. Indien sodanige fout voorkom beïnvloed dit die funksie wat deur die kring uitgevoer word (Yarmolik, 1990:4). Die opsporing van dié tipe fout behels die identifisering van 'n kombinasie van vasgeklem-by-waarde foute, bruggingsfoute sowel as omgekeerde foute.

2.5.2.5 Naburige (neighbour) fout modelle

Hierdie modelle veronderstel dat elke node, net soos by die vasgeklem-by-waarde model, geskakel word vanaf 'n 1 na 'n 0 en vanaf 'n 0 na 'n 1. Die modelle maak verder voorsiening daarvoor dat indien

'n node ge dit nie sal veroorsaak dat naburige nodes geskakel word nie (Hakim, 1989:92).

2.5.3 Sensitiewe-pad begrip

Die belangrikste aspek van toetspatroongenerering is om te verseker dat 'n fout wat by die inset van 'n komponent voorkom, deur die komponent verplaas word, sodat dit tydens toetsing 'n verandering by die uitset van die komponent teweegbring. Hier kom die sensitiewe-pad begrip na vore. As voorbeeld kan na 'n drie-inset EN-hek gekyk word met 'n fout wat op inset C van die hek voorkom.

A	B	C	F
Ø	Ø	Ø	Ø
Ø	Ø	1	Ø
Ø	1	Ø	Ø
Ø	1	1	Ø
1	Ø	Ø	Ø
1	Ø	1	Ø
1	1	Ø	Ø
1	1	1	1

Kondisies vir toetsing van node C.

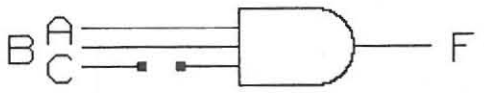


Fig. 2.1: Drie-inset EN-hek

Aanvaar C is oopkring en dat die hek dit as 'n 1 aanvaar. Om die fouttoestand by die uitset van die hek waar te neem, moet die waardes op die oorblywende insette (A en B) so gekies word dat die fout na die uitset van die hek verplaas word. Aangesien die hek onder bespreking 'n EN-hek is, is dit duidelik dat insette A en B albei 1 moet wees (A=B=1). Die uitset van

'n normale hek ander in ooreenstemming met die inset wat op C gekoppel word. 'n Sensitiewe-pad word dus geskep vir die verplasing van die fout vanaf die foutiewe node (inset C) na die uitset van die hek. Indien A, B of albei egter 'n 0 gemaak word, sal die uitset 'n 0 wees ongeag die logikavlak op C.

Dit is belangrik om daarop te let dat daar vir enige hek 'n sensitiewe-pad geskep kan word behalwe vir die eksklusiewe OF-hek.

2.5.4 Beheerbaarheid en waarneembaarheid

Volgens 2.5.3 is 'n belangrike aspek in die diagnosering van 'n fout die vereiste dat 'n fout vanaf die foutiewe node na die uitset van die kring verplaas word. 'n Probleem ontstaan hiermee indien die komponent diep (nie as primêre inset of -uitset nie) in 'n komplekse kring voorkom. Die insetvektore moet nog steeds die foutiewe komponent bereik en die foutiewe toestand moet na die uitset van die kring verplaas word waar dit deur die OTS gelees kan word. Dit word vereis aangesien die primêre insette en -uitsette van die kring die enigste verbinding is tussen die EOT en die OTS.

Daar bestaan dus twee stappe indien enige node getoets word. Eerstens moet die insetwaardes so gekies word dat dit die betrokke node wat getoets word aktiveer. Dit staan as beheerbaarheid bekend. Tweedens moet die effek

hiervan na die u e EOT verplaas word. Dit staan as waarneembaarheid bekend (Fujiwara, 1990:762).

Die terme beheerbaarheid en waarneembaarheid word by die ontwerp-om-te-kan-toets beginsel gebruik en word verder bespreek onder punt 3.2.1.2.

2.6 Strategie vir die ontwikkeling van 'n toetsstel

Wilkens (1986:27) stel die basiese strategie vir strukturele toetspatroongenerering deur die volgende vloeidiagram voor:

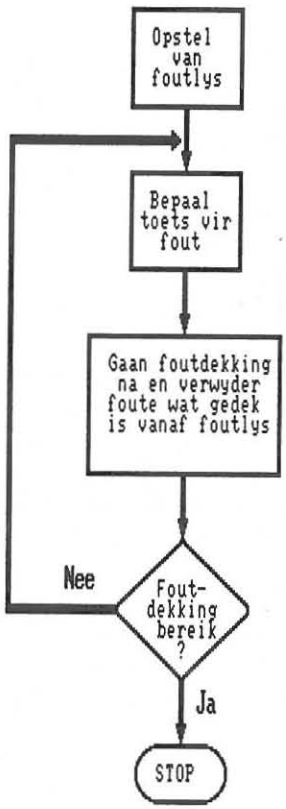


Fig. 2.2: Vloeidiagram vir strukturele toetspatroongenerering.

Aanvanklik word 'n lys saamgestel van al die moontlike foute waarvoor getoets gaan word. Enige foutmodel soos bespreek kan

gebruik word vir [illegible] van die foutlys. Die mees algemene is egter die vasgeklem-by-waarde model.

Vir elke fout wat op die foutlys voorkom word daar 'n toetsvektor bepaal. Alhoewel die toetsvektor bepaal word met die doel om vir 'n enkele fout te toets, is dit tipies dat bykans elke toets ook vir ander foute op die foutlys sal toets.

Ter versekering dat duplisering nie voorkom nie, moet die foutdekking van elke toets bepaal word. Laastens word vasgestel of die verwagte foutdekking al bereik is al dan nie. Indien nie, word die toetsstel verder uitgebrei.

Die foutdekkingsteiken wat gestel word sal afhang van die grootte sowel as die belangrikheid van die kring wat getoets word. Foutdekking gee 'n aanduiding van die aantal foute wat deur toetsvektore geïdentifiseer word en word as 'n persentasie uitgedruk (Seth, Agrawal en Farhat, 1990:582). Foutdekking kan soos volg bereken word:

$$\text{Foutdekking} = \frac{\text{geïdentifiseerde foute}}{\text{moontlike aantal foute - oortollige foute}}$$

Oortollige foute is per definisie foute wat deur geen toets geïdentifiseer kan word nie. Daar moet gepoog word om die foutdekking so na as moontlik aan 100% te kry.

Ter illustrering van bostaande word die kring in figuur 2.3 vervolgens oorweeg.

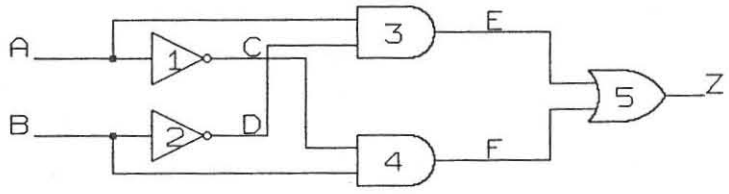


Fig. 2.3: Eksklusiewe OF-hek d.m.v. EN, OF en NIE-hekke

2.6.1.1 Opstel van foutlys

Aangesien daar bewys is dat dit die beste praktiese model is, en terselfdertyd beheerbaar is in terme van kompleksiteit (Hakim, 1989:93), word gebruik gemaak van die vasgeklem-by-waarde model vir die voorbereiding van die foutlys. Die foute wat kan voorkom, en wat ondersoek gaan word, verteenwoordig elke node in die kring vasgeklem-by-0 en vasgeklem-by-1. Met verwysing na figuur 2.3 kan die foutlys as volg saamgestel word:

A/0 ; A/1 ; B/0 ; B/1 ; C/0 ; C/1 ; D/0 ; D/1 ;
E/0 ; E/1 ; F/0 ; F/1 ; Z/0 ; Z/1

A/0 verwys na node A vasgeklem by 0. Dit kan ook geskryf word as A v-b-0 (A vasgeklem-by-0).

2.6.1.2 Bepaling van toets vir moontlike fout

'n Toets word vervolgens vir elke fout bepaal. Daar moet op gelet word wat die logikatoestand by 'n node moet wees om vir 'n fout te kan toets. Indien ons A/0 wil toets, moet die waarde by A gelyk aan 1 gemaak word. Dit word gedoen om onderskeid te tref tussen 'n foutvrye kring en 'n foutiewe kring.

Ten einde te monitor of die fout wel voorkom moet 'n sensitiewe-pad verskaf word vanaf die foutiewe node (A) na die uitset van die kring (Z), sodat die uitset deur die OTS gelees kan word. Om die sensitiewe-pad te stel moet die volgende metode gevolg word:

i) Om hek 3 te aktiveer moet D=1 wees (aangesien dit 'n EN-hek is) en F=0 om hek 5 te aktiveer (aangesien dit 'n OF-hek is).

ii) Om F=0 te kry moet een of albei die insette van hek 4 gelyk aan 0 gemaak word. Vir die foutkondisie is A=1 en dus sal C=0 wees.

Om hek 3 te aktiveer moet D=1 wees en dus sal B=0 wees. Die vektor om A/0 te toets sal dus 10/1 wees of te wel A=1, B=0 en Z=1. Die aanduiding is in die vorm AB/Z, waar die waarde na die streep die foutvrye uitset van die kring aandui. Hierdie toets kan as volg beskryf word:

Die OTS poog om A hoog te skakel. Indien dit egter by 0 vasgeklem is, sal dit deur hierdie toets bepaal kan word aangesien die 0 by A, saam

met die 3, die uitset verkeerdelik na 0 dryf.

Dié pad is egter nie die enigste sensitiewe-pad wat vir die fout geskep kan word nie - daar kan ook 'n pad deur hekke 4 en 5 geskep word. Vir 'n sensitiewe-pad deur hekke 4 en 5 moet die volgende metode gevolg word:

i) Om hek 4 te aktiveer moet B=1 wees en om hek 5 te aktiveer moet E=0 wees. Met B=1 sal D=0 wees en dus sal die kondisie E=0 bevredig word.

Aangesien A=1 is, sal C=0 wees en dus is F=0. Die foutvrye uitset sal Z=0 wees en die vektor kan dus as 11/0 geskryf word. Daar bestaan dus twee toetse vir A/0 naamlik 10/1 en 11/0.

2.6.1.3 Bepaling van die foutdekking

Die toets wat vir A/0 uitgewerk is kan as volg gekontroleer word: Neem die inset van die eerste vektor, naamlik 10, en voer dit by die inset van die kring in. Teken nou die stand van elke node in die kring aan soos in figuur 2.4.

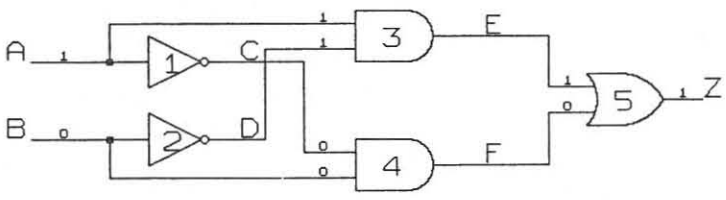


Fig. 2.4: Kondisie van elke node met inset A=1 en B=0

Om nou te A/0 wel geïdentifiseer sal word, word A na 'n 0 verander (foutkondisie). Oorweeg opnuut die uitset van die kring - dit verander vanaf 1 na 0. Aangesien die fouttoestand op die inset van die kring 'n verandering in die foutvrye uitset teweeggebring het kan aanvaar word dat die toets A=1 en B=0 wel gebruik kan word om vir die fout A/0 te toets. Hierdie beginsel is 'n belangrike kenmerk van toetsvektore - indien daar 'n fout in die kring voorkom sal die betrokke vektor die fout identifiseer indien 'n uitset anders as die foutvrye toestand van die kring voorkom.

Op dieselfde wyse kan bewys word dat B/1, D/0, E/0 en Z/0 ook deur dieselfde toets getoets sal word aangesien al hierdie foute 'n foutiewe uitset sal lewer. Die foutdekking van 11/0 kan op dieselfde metode bepaal word as A/0, B/0, C/1, D/1, E/1, F/1 en Z/1.

Daar bestaan dus twee toetse vir die toetsing van A/0. Dit gebeur egter soms dat net een toets bestaan vir 'n bepaalde fout. Na die toets word verwys as 'n essensiële-toets. Alle essensiële-toetse moet in die toetsstel ingesluit word ten einde 'n kring ten volle te toets.

2.6.1.4 Divergensie en rekonvergensie

'n Verskynsel wat algemeen in digitale kringe voorkom is divergensie. Dit is die effek wanneer 'n enkel node vertak in twee of meer nodes. Indien die vertakkings weer later in die kring kombineer, staan dit as rekonvergensie bekend en kan dit probleme veroorsaak ten opsigte van die toetsing van die kring. Rekonvergensie kan onderverdeel word in positiewe- en negatiewe-rekonvergensie. Ter illustrering hiervan word na figuur 2.5 verwys.

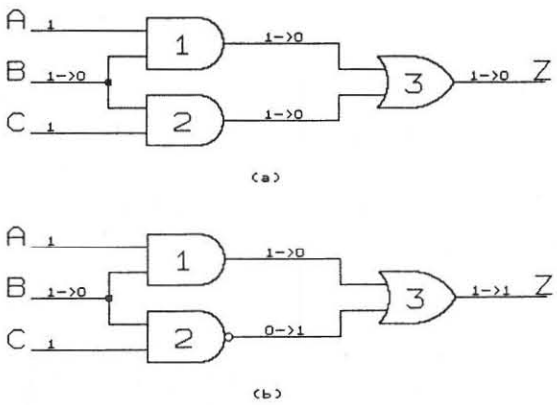


Fig. 2.5: Rekonvergensie, (a) positief en (b) negatief

Beskou die geval waar 'n insetwaarde van 111 op figuur 2.5(a) aangelê, en node B verkeerdelik vasgeklem word by 0. Soos aangetoon, word die fout deur beide hekke 1 en 2 voortgeplant as gevolg van die waardes op A en C. Aangesien die fout 'n ongewensde verandering in die uitset van 1 na 0 veroorsaak, kan die fout wel geïdentifiseer word. Dit staan as positiewe-rekonvergensie bekend.

Indien die [redacted]er word soos in figuur 2.5(b) ontstaan daar 'n probleem. Aangesien die inset van hek drie vanaf 10 na 01 verander, bly die uitset van hek drie onveranderd. Die geval staan as negatiewe-rekonvergensie bekend. Positiewe-rekonvergensie verskaf geen probleme indien 'n kring getoets word nie, terwyl negatiewe-rekonvergensie veroorsaak dat 'n kring dikwels nie volkome toetsbaar is nie.

2.6.1.5 Ontoetsbare foute

Tydens die bepaling van toetsvektore van 'n kring ontstaan daar soms konfliksituasies. Ter illustrering hiervan dien figuur 2.6.

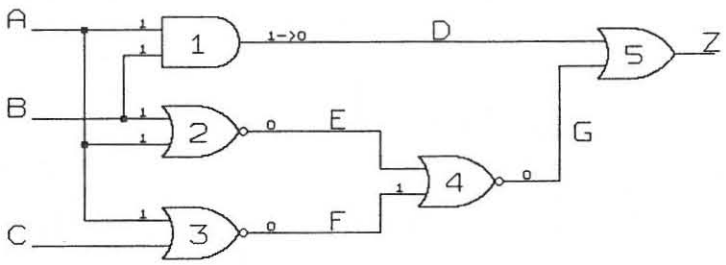


Fig. 2.6: Kring met ontoetsbare fout

Oorweeg die toetsing van D/0. Hiervoor word D=1 gestel om die fout te kan genereer. Die volgende stappe moet gevolg word om die sensitiewe-pad te stel:

i) Ten einde D=1 te kry moet A=B=1 wees. Dit veroorsaak dat die uitset van hek 2 gelyk is aan 0.

ii) Vir 'n sensitiewe-pad deur hek 5 moet G=0 wees. Aangesien die uitset van hek 2 reeds 0 is, moet F=1 wees.

iii) Aangesien die een inset van hek 3 reeds 1 is (A=1), is F=0.

'n Konfliksituasie bestaan nou aangesien F terselfdertyd by 0 en 1 moet wees. Dit is uiteraard onmoontlik en die fout D/0 is dus ontoetsbaar.

Ontoetsbare foute word gewoonlik veroorsaak deur oortollige komponente in die kring. Die funksie van die kring word voorgestel deur:

$$Z = \overline{\overline{(A \cdot B)} + \overline{(A+B)} + \overline{(A+C)}}$$

$$= (A \cdot B) + (A+B)(A+C)$$

$$= A \cdot B + A + A \cdot C + A \cdot B + B \cdot C$$

$$= A + B \cdot C$$

By nadere ondersoek word gevind die uitset van hek 4 is:

$$G = \overline{\overline{(A+B)} + \overline{(A+C)}}$$

$$= (A+B)(A+C)$$

$$= A + A \cdot C + A \cdot B + B \cdot C$$

$$= A + B \cdot C$$

Hekke 1 en 5 is dus oortollig en dit veroorsaak die ontoetsbare fout. Indien die kring sonder hekke 1 en 5 ondersoek word, word gevind dat die ontoetsbare foute nie meer voorkom nie en die kring sal ten volle getoets kan word. Soms word kringe egter so uitgebrei om die voorkoms van gevaarsituasies (hazards) te beperk (Wilkenson, 1987:385).

2.6.2 Toetsing van volgorde kringe

By 'n kombinasiekring word die uitset van elke hek bepaal slegs deur die inset wat op daardie stadium op die inset van die hek toegepas word. By volgordekringe word die resultaat nie net deur die insette bepaal nie, maar ook deur die huidige stand waarin die kring verkeer - met ander woorde, wat het in die verlede met die kring gebeur? Dit kan veroorsaak dat 'n aantal insette opeenvolgens op 'n kring toegepas moet word om vir 'n enkele vasgeklem-by-waarde fout te toets. Een benadering wat gevolg kan word by die toetsing van 'n volgordekring is om die kring so te verander dat dit soos 'n kombinasiekring getoets kan word. Dit verwys na die ontwerp-om-te-kan-toets tegnieke soos beskryf in 3.2.

Boonop kan 'n volgordekring in enige begintoestand wees direk na aanskakeling. Beheer moet uitgeoefen word om die toestand van die kring so te verander dat dit in 'n bekende staat verkeer voordat die toetsing 'n aanvang neem (Capillo, 1990:328).

Alle wipkringe funksioneer nie dieselfde nie. By vlaksensitiewe wipkringe, soos die D-tipe houkring, reageer die uitset onmiddelik volgens die inset solank die klok aktief is. Indien die klok onaktief is bly die uitset stabiel. By randgesnellerde wipkringe reageer die uitset slegs gedurende die oomblik wanneer die klok van staat verander. By JK meester/slaaf wipkringe kan 'n gevaarsituasie wat voorkom op die J of K inset gedurende

die positiewe deel van die klok onderhou word, wat veroorsaak dat die wipkring stel of herstel indien die klok verander na 'n LAAG.

Wilkens (1986:76) beskou die ontwikkeling van 'n toetsstel vir volgordekringe meer as 'n kuns as wat dit 'n wetenskap is.

2.6.2.1 Opstel van 'n foutlys vir volgorde kringe

Indien die foutlys van 'n wipkring opgestel word, word interne nodes van die wipkring nie oorweeg nie, maar slegs die insette en uitsette van die wipkring. In so 'n geval sal die foutlys bestaan uit die insette, uitsette, klok, stel en herstel leidings van die wipkring vasgeklem-by-0 en vasgeklem-by-1. Indien enige komplekse komponent getoets word, word slegs die funksies wat in die normale werking van die kring gebruik word in die toets ingesluit.

2.6.2.2 Plasing van kring in begintoestand (Initialization)

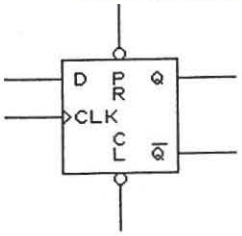
Indien 'n kring nie in 'n bekende begintoestand geskakel kan word nie kan die kring nie getoets word nie (Turino, 1990:36). Onbekende toestande word deur die kring voortgeplant indien die uitset nie bekend is voordat met die toetsing begin word nie. Die eenvoudigste metode om die kring in 'n bekende

toestand te plaas is om gebruik te maak van die stelleiding (PR) en herstelleiding (CL).

Indien dit moeilik is om 'n kring in die begintoestand te skakel sal dit langer toetstye en 'n minder as optimale foutdekking tot gevolg hê. Probleme kan ook ondervind word met die opstel van sodanige toetsvektore. Indien toetsvektore vir 'n volgordekring opgestel word, word die kring 'n aantal kere deur die loop van die toetsing in die begintoestand geskakel. Die stel- en herstel-leidings moet beskikbaar wees vir toetsdoeleindes, selfs indien dit nie in die normale werking van die kring gebruik word nie.

2.6.2.3 Opstel van toetsvektore vir volgorde kringe

Die asinkrone insette van 'n wipkring behoort eers getoets te word voordat dit gebruik word om die kring in 'n begintoestand te plaas. Die stel- en herstelleidings van die wipkring word as asinkrone insette gedefinieer aangesien die uitset slegs deur die huidige insette bepaal word en nie deur enige vorige kondisies van die kring nie. Die wipkring reageer dus net soos 'n gewone hek ten opsigte van asinkrone insette en nie soos 'n volgorde kring nie. Ter illustrering van die toetsing van die asinkrone insette word die D-tipe wipkring (7474) oorweeg:



D	Q
Ø	1
1	Ø

Fig. 2.7: D-tipe wipkring (7474)

Enige van die asinkrone insette kan eerste getoets word. Tabel 2.1 toon die verlangde toetsvektore vir genoemde wipkring.

Soos in die tabel gesien kan word sal die toetse wat vir die asinkrone insette geskryf word ook die uitset foute van die wipkring dek. Die moontlikheid bestaan dat Q=1 is met die aanskakeling van die kring. Daar kan dus nie aanvaar word dat toets 1 wel PR/1 toets nie en daarom word toets 3 benodig.

Tabel 2.1: Toetsvektore vir D-tipe wipkring (D=CLK=1)

Toets	Insette	Foutvrye-uitsette	Foutdekking
1	PR=0;CL=1	Q=1;$\overline{Q}$=0	Q/0;$\overline{Q}$/1;CL/0
2	PR=1;CL=0	Q=0;$\overline{Q}$=1	Q/1;$\overline{Q}$/0;PR/0;CL/1
3	PR=0;CL=1	Q=1;$\overline{Q}$=0	Q/0;$\overline{Q}$/1;PR/1;CL/0

Die keuse van watter fout eerste voor getoets moet word, moet sorgvuldig uitgeoefen word aangesien 'n verkeerde keuse sal veroorsaak dat 'n aantal ekstra vektore benodig sal word om die wipkring in die korrekte staat te plaas. Ten einde die sinkrone

werking van die wipkring te evalueer word PR en CL by hul onaktiewe vlakke gehou, naamlik PR=CL=1. Om die D inset te toets word die wipkring eers in 'n beginstaat geplaas deur PR LAAG te puls wat die wipkring sal stel. Die wipkring is nou in 'n bekende staat en daar kan voortgegaan word met die toetsing. Die inset op D word nou so gekies dat, indien 'n klok toegepas word op die wipkring, daar 'n verandering in die uitset van die wipkring voorkom.

Daar word begin deur vir D/1 te toets - en dus word D=0 gestel. Nadat 'n positiefgaande klokpuls toegedien is sal Q=0 wees en NQ=1. Dit is die foutvrye uitset van die wipkring. Daar sal nie net vir D/1 getoets word nie, maar CLK/0, Q/1 en NQ/0 sal ook geïdentifiseer word. Om D/0 te toets word D=1 gestel en 'n klokpuls toegepas.

Aangesien die klok toegepas word nadat die insette aangelê is sal tydafhanklike probleme nie ondervind word nie. Die OTS kan dit identifiseer indien CLK vasgeklem word by 1 of by 0.

2.6.3 Toetsing van bruggingsfoute

Soms is die enkel vasgeklem-by-waarde foutmodel nie voldoende vir die toetsing van 'n kring nie. 'n Model wat dikwels bygevoeg word om 'n kring meer volledig te toets is die bruggingsfout model.

2.6.3.1 Opste_ van .. foutlys

Soos in 2.5.2.2 genoem, word die gedrukte stroombaan uitleg eers afgehandel voordat besluit word tussen watter bane vir bruggingsfoute getoets moet word. Die foutlys sal dus deur die toetsingenieur saamgestel word deur aangrensende bane of penne van komponente in ag te neem. Die foutdekking van die kring sal beïnvloed word deur die hoeveelheid bruggingsfoute wat in die foutlys ingesluit word - hoe meer bruggingsfoute daar voor getoets word, hoe groter is die foutdekking.

2.6.3.2 Opstel van toetsvektore vir bruggingsfoute

Ter illustrering van die metode wat gevolg word wanneer vir 'n bruggingsfout in 'n kring getoets word, word daar na figuur 2.8 verwys. In dié kring word 'n bruggingsfout tussen nodes C en D voorgestel.

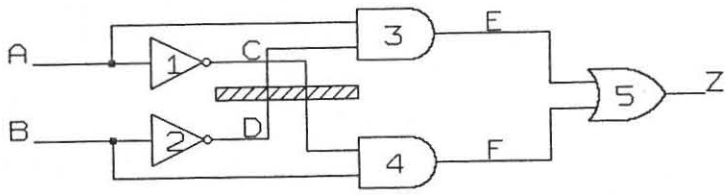


Fig. 2.8: Bruggingsfout tussen nodes C en D.

C moet by 'n ander logikatoestand as D wees ten einde te kan toets vir 'n bruggingsfout tussen genoemde twee nodes in figuur 2.8. Dit kan gedoen

word deur C=1 en D=0 te stel, of deur C=0 en D=1 te stel. Indien veronderstel word dat van TTL-hekke gebruik gemaak word sal die LAAG (0) oorheers. Die node wat by 1 is sal dus afgetrek word na 0 indien 'n bruggingsfout wel voorkom.

As voorbeeld word C=1 en D=0 gestel en, as gevolg van die defek in die kring, sal C verander vanaf 1 na 'n 0. Om die fout deur hek 4 te stuur moet B=1 wees wat reeds waar is aangesien D=0 is. Om C=1 te kry moet A=0 gestel word. Om die sensitiewe-pad na die uitset van die kring te voltooi, moet E=0 wees om die fout deur hek 5 te stuur. Dit word wel verkry aangesien A=D=0 is. Die uitset van die kring is logika 0, anders as die verwagte 1, wat die teenwoordigheid van die bruggingsfout bewys.

Sommige bruggingsfoute sal geen foutiewe uitset van die kring teweegbring nie en kan dus nie voor getoets word nie.

2.6.3.3 Bepaling van die foutdekking

Aangesien 'n toetsvektor vir elke bruggingsfout opgestel word, sal die foutdekking voltooi wees wanneer die toetsingenieur tevrede is met die aantal bruggingsfoute, soos bepaal volgens die gedrukte stroombaan uitleg, waarvoor getoets is.

2.7 Foutineenstorting

In enige kring is die aantal foute, wat deur die enkel vasgeklem-by-waarde foutmodel voorgestel word, gelyk aan 2n, waar n die aantal nodes is wat in die kring voorkom. 'n Kring van redelike grootte kan dus 'n geweldige hoeveelheid foute hê, en dus word baie berekenings vereis tydens die opstel van die toetsvektore. Die grootte van die foutlys kan egter verklein word deur middel van 'n proses wat as foutineenstorting bekend staan (Wilkens, 1986:45).

Ter verduideliking van foutineenstorting word verwys na tabel 2.2. Die matriks toon foute f1 tot f4 op die horisontale-as aan en die toetse wat benodig word om vir die foute te toets op die vertikale-as. Die kruisies (X) dui aan watter foute deur watter toetse gedek word.

Foutineenstorting bestaan uit twee aksies, naamlik die identifisering van ononderskeidbare- en dominantefoute.

Tabel 2.2: Foutmatrikspatrone wat foutineenstorting toelaat

	f1	f2	f3	f4
t_1	X	X	X	
t_2			X	X
t_3	X	X	X	X

2.7.1 Ononderskeidbarefoute

Uit die tabel kan vir beide f1 en f2 deur toetse t_1 en t_3 getoets word. Dié twee foute staan as

ononderskeidbarefoute bekend en dit is nie nodig dat albei in die foutlys ingesluit word nie. Tydens die samestelling van die foutlys kan byvoorbeeld f1 in die lys ingesluit, en f2 weggelaat word. Die ononderskeidbarefoute kan as {f1,f2} aangedui word. Deur alle sodanige gevalle te identifiseer kan die aantal foute waarvoor getoets word dus verminder word.

2.7.2 Dominantefoute

Die tweede groep foute wat 'n vermindering in die aantal foute in die foutlys tot gevolg kan hê staan as dominantefoute bekend. 'n Voorbeeld hiervan is f3 en f4. Hier kan gesien word dat f4 'n subgroep van f3 vorm. Indien t_2 uitgevoer word sal vir f3 sowel as f4 getoets word. In dié geval kan f3 uit die foutlys weggelaat word. Dit is belangrik om daarop te let dat die fout wat deur die meeste toetse getoets kan word uit die foutlys verwyder word. Dominantefoute kan aangedui word as {f4->f3} waar f3 die fout is wat weggelaat kan word.

2.7.3 Foutineenstorting op TTL-hekke

Die metode wat gebruik is om foutineenstorting vanaf 'n foutmatriks toe te pas, kan nie ten opsigte van komplekse kringe gebruik word nie aangesien dit net vir klein kringe prakties is om 'n foutmatriks op te stel. Indien die gewone toetstegnieke egter op enkel hekke toegepas word, word gevind dat daar ooreenkomste bestaan tussen die ononderskeidbarefoute en die dominantefoute

vir elke tipe hek. Vir standaard hekke met twee insette elk kan tabel 2.3 saamgestel word.

Tabel 2.3: Foutineenstorting vir standaard hekke

Hek	Ononderskeidbarefoute	Dominantefoute
EN-hek	{A/0,B/0,Z/0}	{A/1, B/1 -> Z/1}
OF-hek	{A/1,B/1,Z/1}	{A/0, B/0 -> Z/0}
NEN-hek	{A/0,B/0,Z/1}	{A/1, B/1 -> Z/0}
NOF-hek	{A/1,B/1,Z/0}	{A/0, B/0 -> Z/1}
NIE-HEK	{A/0,Z/1};{A/1,Z/0}	GEEN

Die toestande soos in tabel 2.3 aangegee word kan nou op enige kring toegepas word voordat die foutlys saamgestel word. Hierdie reëls kan egter nie gebruik word op enige nodes wat divergensie bevat nie. Indien 'n hek byvoorbeeld drie insette bevat en slegs een van die insette bevat divergensie, kan die reël egter nog steeds op die oorblywende twee insette en die uitset toegepas word.

2.8 Diagnosering van die foute

Die evaluering van 'n bord eindig nie by die bepaling van die betrokke toetsvektore nie. Informasie moet saam met die toetsstel ingesluit word om die foutiewe node of komponent in die kring te bepaal aan die hand van die resultate wat deur die toetse verkry is. Ter illustrering hiervan word opnuut verwys na figuur 2.3. Foutineenstorting word maksimaal benut deur dit toe te pas voordat die toetsvektore uitgewerk word.

Die moontlike foute wat in die kring kan bestaan kan as volg uiteengesit word:

f0 = Geen fout in kring
f1 = A/0
f2 = A/1
f3 = B/0
f4 = B/1
f5 = {C/0,F/0} (Ononderskeidbaar)
f6 = C/1
f7 = {D/0,E/0} (Ononderskeidbaar)
f8 = D/1
f9 = {E/1,F/1,Z/1} (Ononderskeidbaar)
f10 = Z/0

Die foute, tesame met die minimum toetsstel, word in tabel 2.4 as 'n foutmatriks voorgestel. Die foute word op die horisontale-as voorgestel en die toetse op die vertikale-as.

Tabel 2.4: Foutmatriks van die kring in figuur 2.3

	f1	**f2**	**f3**	**f4**	**f5**	**f6**	**f7**	**f8**	**f9**	**f10**
t_{00}		X		X					X	
t_{01}		X	X		X					X
t_{10}	X			X			X			X
t_{11}	X		X			X		X	X	

Tydens die bepaling van die minimum toetsstel moet sorg gedra word dat ten minste een maal vir elke fout getoets word. Uit die foutmatriks spruit dat f6 en f8 ononderskeidbarefoute is. Addisionele toetse moet sover moontlik bepaal word ten einde

onderskeid te tr die ononderskeidbarefoute. Aangesien die minimum toetsstel in die voorbeeld ook die volledige toetsstel is, bestaan daar geen addisionele toetse nie en daar sal dus nie onderskeid getref kan word tussen f6 en f8 nie.

Vanaf die foutmatriks kan daar 'n metode van besluitneming gebruik word om die foutiewe komponent of groep komponente te identifiseer. Die inligting in tabel 2.4 word gebruik om 'n foutdiagram soos in figuur 2.9 saam te stel. Slegs die toetse wat in tabel 2.4 as die minimum toetsstel gelys is word vir die toetsing gebruik.

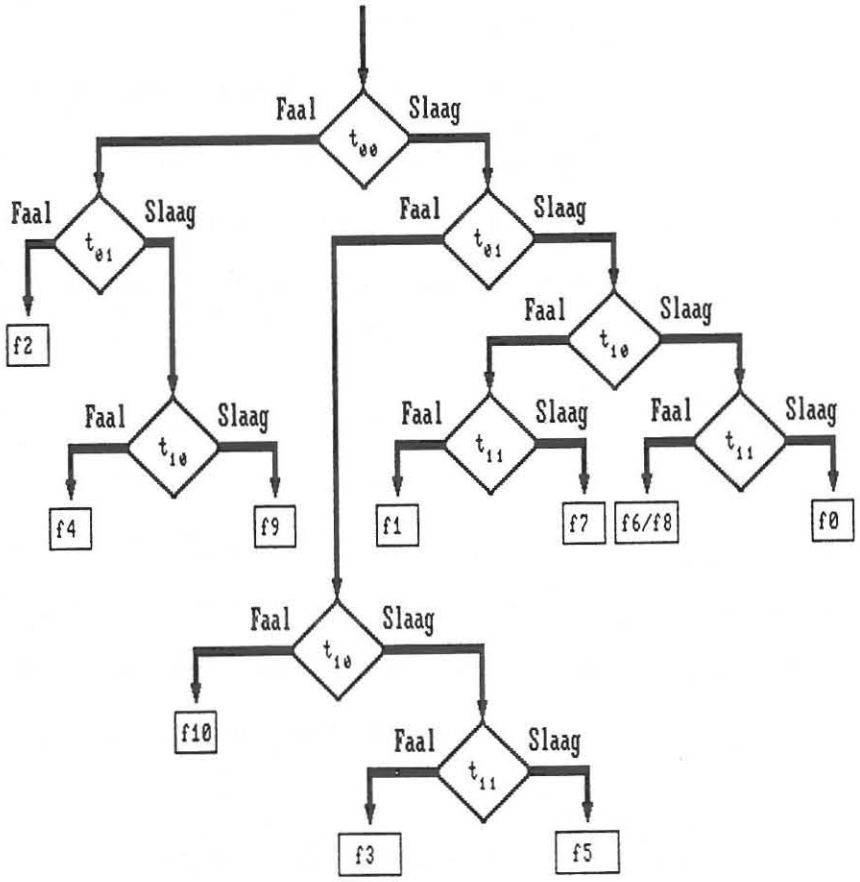


Fig. 2.9: Foutdiagram van figuur 2.3

Die evaluering van die kring kan begin word deur t_{00} uit te voer. Indien 'n foutvrye uitset verkry word, is f2, f4 en f9 as foutvry bewys. Die volgende toetse op die lys kan nou uitgeoefen word soos aangedui en indien al die toetse slaag is die kring foutvry van al die foute wat in die foutlys voorkom. Indien toets t_{00} faal kom een van die foute f2, f4 of f9 wel voor. Om die fout verder te diagnoseer word t_{01} uitgevoer. Indien die fout faal is f2 foutief wat dan beteken dat A vasgeklem is by 1. Indien t_{01} slaag moet verdere toetse uitgevoer word om te bepaal watter van f4 of f9 foutief is. Dieselfde beginsel word gebruik om die res van die foute te diagnoseer.

Die belangrikste voordeel by die gebruik van die foutdiagram is dat toetsing van die bord ten volle outomaties uitgevoer word met geen interaksie van die operateur nie. Daar bestaan egter 'n aantal nadele by die gebruik van die metode, naamlik:

i) Die foutdiagram benodig 'n baie groot databasis. Vir 'n kring met n uitsette het elke toets 2^n moontlike resultate (Wilkens, 1986:58).

ii) Alhoewel die toepassing van die foutdiagram baie goed funksioneer indien 'n kombinasiekring getoets word, werk dit nie baie goed wanneer 'n volgordekring getoets word nie.

iii) Die voorkoms van ononderskeidbarefoute veroorsaak dat die fout soms net tot 'n groep foute gediagnoseer kan word. Nogtans behoort die finale foutopsporing dan fisies maklik gedoen te kan word.

'n Moontlike oplossing indien volgordekringe getoets moet word, of foute tot enkel komponente geïsoleer moet word, is om gebruik te maak van 'n geleidetaster (guided probe) (Nazili, 1988:24). Die geleidetaster verskaf 'n ekstra kanaal na die OTS en kan gebruik word om 'n node HOOG of LAAG te dryf of om inligting vanaf die node te versamel (Fluke, 1987:20). Die gebruik van 'n geleidetaster veroorsaak egter dat die toetstyd baie verleng word. Sorg moet ook gedra word dat 'n node met baie divergensie nie oormatig belas word nie.

2.9 Outomatiese-toetspatroongenerering (OTPG)

Daar bestaan 'n aantal outomatiese-toetspatroongenerators vir digitale kringe waarvan die meeste die vasgeklem-by-waarde foutmodel gebruik en terselfdertyd poog om die fout na die uitset van die kring te stuur (Randell en Treleaven, 1983:180). Weens die eksponensiële toename van probleme moet OTPG vir baiegrootskaalse-integrasie as onmoontlik bestempel word (Randell en Treleaven, 1983:180). 'n Verskeidenheid van data word benodig vir OTPG. Dit sluit onder meer 'n netdiagram, wat die komponente en hulle onderskeie verbindings aantoon, in. 'n Verdere lys word benodig van alle komponente met hulle spesifikasies, soos dit op die vervaardiger se inligtingsblaaie verskyn. 'n Foutlys word ook benodig vir die kring wat getoets moet word.

'n Hele aantal toetspatroongenerator algoritmes is al bekend gestel waarvan die belangrikste die D-Algoritme, PODEM ("Path-Oriented Decision Making") en FAN is (Fujiwara en Shimono, 1983:1137).

2.10 Opsomming

Toetsing van 'n kring behels die herhaaldelike toepassing van logikawaardes by die primêre insette van die kring en die waarneming van die resultate by die primêre uitsette van die kring. Hierdie toetse kan outomaties uitgevoer word deur gebruik te maak van 'n outomatiese-toetsstelsel (OTS).

Die samestelling van 'n toetsstel behels die toepassing van foutineenstorting op die kring, die bepaling van 'n foutlys, die skryf van toetsvektore volgens die foutlys en die bepaling van die foutdekking. Vir die diagnosering van die fout word 'n foutmatriks opgestel waarvanaf die foutiewe node of komponent aan die hand van die resultate bepaal word.

Die samestelling van 'n toetsstel vir die diagnosering van 'n fout is 'n lang en ingewikkelde proses. Ontwerp-om-te-kan-toets tegnieke kan gebruik word om hierdie proses te vergemaklik.

ONTWERP-OM-TE-KAN-TOETS

3.1 Hoekom ontwerp-om-te-kan-toets?

Die kompleksiteit van digitale kringe veroorsaak dat die ontwikkeling van 'n toetsstel nie in 'n redelike tyd of teen lae koste ontwikkel kan word nie. Deur die ontwerp van 'n kring te verander kan toetsing - en dus die ontwikkeling van 'n geskikte toetsstel - vergemaklik word. Ten spyte van die ekstra tyd wat spandeer word om 'n makliker toetsbare ontwerp te verseker, sowel as die finansiering van addisionele komponente as gevolg van so 'n ontwerp, verseker so benadering dat die totale koste van die produk uiteindelik laer is (Turino, 1990:13).

3.2 Ontwerp-om-te-kan-toets tegnieke

Die metodes wat gebruik word om toetsbare kringe te ontwerp kan in drie kategorieë verdeel word, naamlik Ad hoc, struktureel en selftoetsing.

3.2.1 Ad hoc tegnieke

Ad hoc tegnieke behels die ontwerp van 'n kring sodat toetspatroongenerering ekonomies uitvoerbaar is (McCluskey en Bozorgui-Nesbat, 1981:866). Die volgende praktiese ontwerptegnieke kan vir kombinasie- sowel as

volgordekringe [illegible] ten einde toetsbaarheid te verseker.

3.2.1.1 Opdeling van die kring

'n Belangrike tegniek is die opdeling van 'n kring in kleiner eenhede, wat dan elk op sy eie makliker getoets kan word. Vir die opdeling van 'n kring kan van multipleksers of hekke gebruik gemaak word. Die beginsel word verduidelik aan die hand van figuur 3.1.

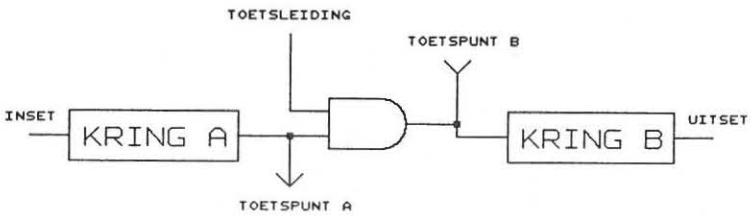


Fig. 3.1: Opdeling van 'n kring met behulp van EN-hek

Die oorspronklike kring word verdeel in kringe A en B deur middel van 'n EN-hek. Gedurende die normale werking van die kring word die toetsleiding HOOG gehou terwyl dit LAAG gemaak word gedurende die toetsing van die kring. Gedurende toetsing word toetspunte A en B onderskeidelik as uitset vir kring A en inset vir kring B gebruik. Indien 'n kring verdeel word moet komponente wat 'n logiese eenheid vorm sover moontlik as 'n eenheid behou word.

3.2.1.2 Addisionele toetspunte

'n Eenvoudige metode om die toetsbaarheid van 'n kring te verhoog is deur die byvoeging van addisionele toetspunte (Turino, 1990:52). 'n Goeie posisie vir die plasing van addisionele toetspunte in 'n kring wat uit 'n aantal blokke in serie bestaan - en wat goeie resultate lewer - is in die middel van die kring. So 'n toetspunt verminder die toetskoste aansienlik. Addisionele toetspunte kan weer in die helftes wat so gevorm is geplaas word. Die verwagte aantal toetse wat uitgevoer word op 'n kring met 'n aantal blokke in serie is as volg (Loveday, 1980:169):

i) uitset-na-inset/inset-na-uitset

$$C = 1/2n(n-1)(n+2)$$

ii) half-opdeel

$$C = 3,32 \log n$$

waar n = aantal blokke in serie

Met 10 blokke in serie lewer die half-opdeel tegniek 3,32 verwagte toetse teenoor 5,4 vir die uitset-na-inset.

Die beheerbaarheid en waarneembaarheid van elke node is, soos in 2.5.4 bespreek, van kardinale belang. Indien 'n digitale kring ontwerp word moet voorsiening gemaak word dat beheer uitgeoefen kan word op die insette van komponente soos wipkringe, tellers en die adres insette van multipleksers en demultipleksers. Voorsiening moet ook gemaak word om

die uitsette van die genoemde komponente te kan waarneem. Om hieraan te voldoen kan van addisionele toetspunte gebruik gemaak word.

Die addisionele toetspunte soos bespreek word aan die randkonnekteerder verbind waar dit toeganklik is vir die OTS.

3.2.1.3 Terugvoerlusse

Terugvoerlusse het tot gevolg dat die diagnosering van 'n fout dikwels nie tot 'n enkel komponent beperk word nie. Die uitset van die lus, wat foutief is, word na die inset teruggevoer, en so word die fout deur die lus voortgeplant. Die diagnosering van die fout word beperk tot die identifisering van al die komponente in die lus en dus moet daar gepoog word om die terugvoerlus te breek tydens toetsing.

'n Terugvoerlus kan verbreek word deur gebruik te maak van 'n fisiese skakelaar in die kring of deur middel van 'n EN-hek, soos aangetoon in 3.2.1.1.

3.2.1.4 Klokpulsgenerator

'n Probleem ontstaan indien 'n klokpulsgenerator in die middel van 'n kring voorkom en daar geen beheer is oor die werking van die klok nie. Indien die interne klokspoed van die EOT hoër is as die spoed van die OTS, is daar geen metode hoe die uitset van

die EOT kor n word nie. Die uitset van die klok moet aan 'n eksterne punt verbind word waar dit met eksterne meetinstrumente geverifieer kan word. Die klok van die OTS word gebruik vir die toetsing van die kring.

3.2.1.5 Bedraad-EN en bedraad-OF funksies

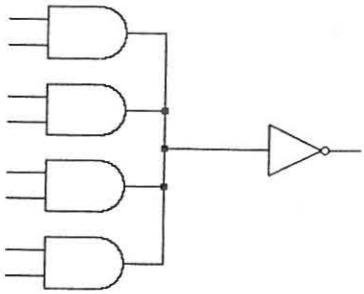


Fig. 3.2: Bedraad-EN funksie (foutiewe ontwerp)

Die gebruik van 'n bedraad-EN (of bedraad-OF) funksie soos aangetoon in figuur 3.2 bemoeilik die diagnosering van 'n fout en moet derhalwe vermy word. Indien 'n fout op die node met konvergensie voorkom kan die foutiewe hek nie bepaal word nie. Om diagnosering van die fout te verbeter word die kring verander soos in figuur 3.3. aangetoon.

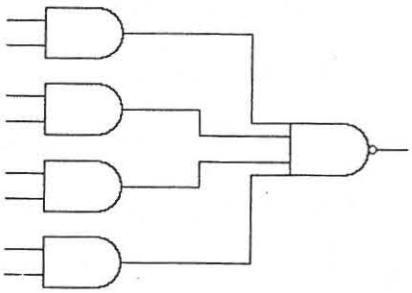


Fig. 3.3: Bedraad-EN funksie (korrekte ontwerp)

3.2.1.6 Tellers en skuifregisters

Indien daar van lang tellers of skuifregisters gebruik gemaak word, moet dit opgebreek word in kleiner dele om die toetsing te vergemaklik. Turino (1990:57) beveel aan dat 'n teller uit 'n maksimum van vier stadiums bestaan.

Neem as voorbeeld 'n 36-bis teller wat getoets moet word. Om die teller ten volle te toets word daar $2^{36}+1=6,87X10^{10}$ klokpulse benodig. Indien die teller in nege 4-bis tellers opgebreek word, word die aantal klokpulse wat benodig word verminder na $9(2^4+1)=153$.

3.2.1.7 Monostabiele multivibrator

Enige tipe asinkrone kring moet sover moontlik vermy word (Ginsberg, 1991:46). Monostabiele multivibrators is moeilik om te toets as gevolg van die tydafhanklike eienskappe van die komponente.

Om die monostabiele multivibrator toetsbaar te maak moet die uitset as 'n toetspunt beskikbaar gestel word vir meetdoeleindes. Beheerleidings moet verskaf word vir die beheer van logika wat normaalweg deur 'n monostabiele multivibrator aangedryf word. Indien twee of meer monostabiele multivibrators, wat in kaskade verbind is, getoets word, moet die verbindings tussen die wipkringe verbreek, en aan

die randkonnekteerder verbind word - vanwaar beheer en waarneming op elke wipkring uitgeoefen kan word.

3.2.1.8 Analoog- en digitale komponente

Toetsvektore vir analoog komponente verskil van dié van digitale komponente. Om toetsing te vergemaklik moet analoog komponente geskei word van digitale komponente.

3.2.1.9 Divergensie en rekonvergensie

By divergensie en rekonvergensie moet daar gesorg word dat die maksimum belasting vir die hekke nie oorskry word nie - selfs nie indien 'n toetspen aan die betrokke node verbind word nie.

3.2.1.10 Oortollige komponente

Oortollige komponente word in 'n kring gebruik vir 'n verskeidenheid redes - soos onder andere die voorkoming van gevaarsituasies (hazards), standaardisering van geïntegreerde stroombane en vir die korrekte werking van 'n kring tydens die voorkoms van foute. Die gebruik van oortollige komponente veroorsaak normaalweg dat 'n kring nie ten volle getoets kan word nie (Sutton en Bredeson, 1980:648). Indien moontlik moet oortollige komponente derhalwe vermy word.

13/03/29

3.2.2 Strukturele benadering

Die tegnieke vir toetspatroongenerering, tesame met die Ad hoc ontwerptegnieke soos bespreek, is gewoonlik voldoende vir die ontwikkeling van 'n toetsstel vir kombinasiekringe. Om probleme by die opstel van 'n toetsstel vir volgordekringe te oorkom, veral by baie-grootskaalse-integrasie, kan egter van addisionele tegnieke gebruik gemaak word.

Dié tegnieke berus op die stelling dat indien alle houkringe tot enige waarde gestel kan word, en indien die uitset direk gelees kan word, dan word toetsgenerering - en moontlike fout isolering - beperk tot die generering van 'n toetsstel soortgelyk aan dié van 'n kombinasiekring (Hakim, 1989:101). 'n Aantal van die tegnieke word kortliks bespreek.

3.2.2.1 Vlak-sensitiewe-aftas-ontwerp (VSAO)

Vlak-sensitiewe-aftas-ontwerp is ontwikkel om die toetsing van komplekse volgordekringe moontlik te maak, sonder die gebruik van baie beheerinsette en waarnemingsuitsette. Aftas (scan) verwys na die vermoë om enige logikatoestand vir toetsdoeleindes in of uit 'n kring te skuif.

Die belangrikste element wat in VSAO gebruik word is die skuifregisterhoukring of SRH (shift register latch). Die algemene struktuur van die SRH word in

figuur 3.4 aangetoon. Hierdie aanpassing van die kring vergemaklik toetsing.

Hierdie kring is bestand teen die meeste wisselstroom onreëlmatighede wat in die klok mag voorkom. Dit vereis slegs dat dit lank genoeg HOOG bly om 'n terugvoerlus te stabiliseer voordat dit na die LAAG staat terugkeer (Hakim, 1989:102).

DATA-IN (DI), DATAKLOK en DATA-UIT (DU) verrig die normale geheuefunksies terwyl AFTASDATA-IN (ADI), AFTASKLOK 1 en 2 en AFTASDATA-UIT (ADU) bykomende leidings is vir die skuifregister funksies. Deur gebruik te maak van addisionele komponente kan enige node tot enige waarde gestel word, of die staat van enige node waargeneem word, sonder die gebruik van 'n klok. Deur AFTASKLOK 1 en 2 tussen die logikavlakke 0 en 1 te skakel kan die logikastaat by enige node, na enige waarde (0 of 1) geskakel word.

Die algemene toepassing van die SRH word in figuur 3.5 aangetoon. 'n Skuifregister word gevorm deur AFTASDATA-UIT van een SRH na AFTASDATA-IN van die volgende te verbind. Om toetsvektore vir die kombinasiekring te bepaal word die waardes vir die primêre insette en die H2 houkringe bepaal. Die

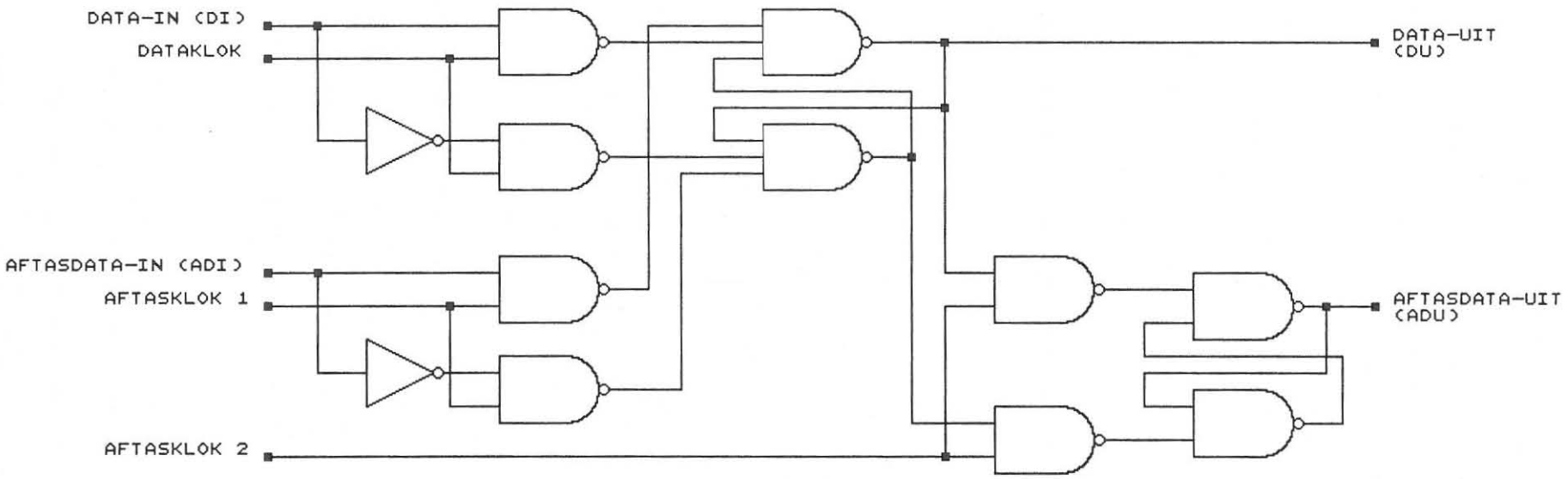


Fig. 3.4: Skuifregisterhoukring (SRH)

waardes wat houkringe benodig word, word in serie deur die kring geskuif vanaf AFTASDATA-IN.

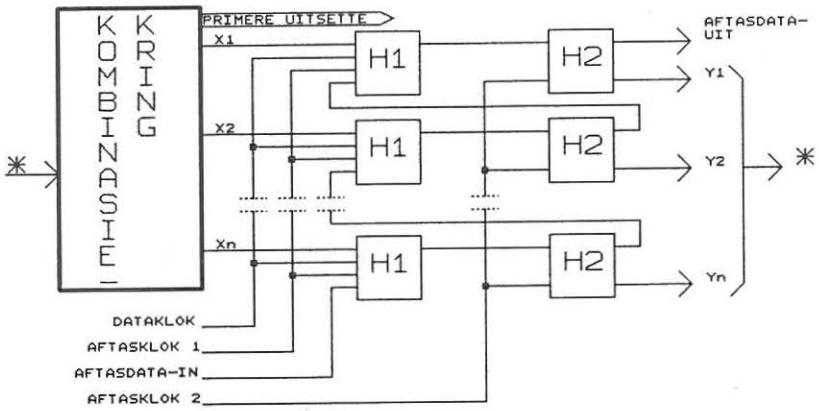


Fig. 3.5: Algemene struktuur van VSAO

Die toetsresultate wat op uitsette X_1, X_2, ..., X_n verkry word kan gelees word deur dit seriaal deur die skuifregister te skuif.

'n Nadeel van VSAO is die seriewerking van die skuifregisters. Die gebruik van VSAO is ook kompleks as gevolg van die byvoeging van ekstra houkringe en insette.

3.2.2.2 Aftasbaan (Scan path)

Die aftasbaan tegniek stem baie ooreen met VSAO. Die geheue-element wat gebruik word, word in figuur 3.6. aangetoon.

Gedurende die normale werking van die kring word AFTASKLOK 2 HOOG gehou. Dit verseker dat AFTASDATA-IN geen invloed op houkring 1 het nie. Die waardes

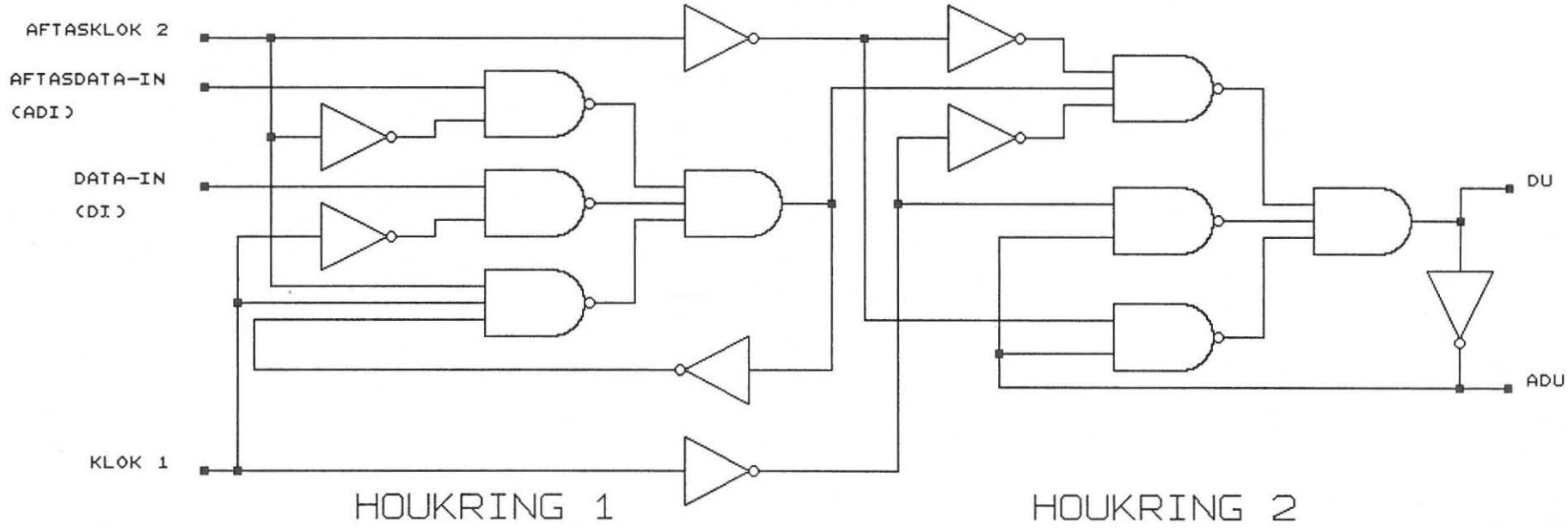


Fig. 3.6: Aftasbaan tegniek

in houkrin onveranderd. Indien KLOK 1 LAAG skakel word data in houkring 1 geplaas en as KLOK 1 terugskakel na HOOG word die uitset van houkring 1 na houkring 2 oorgeplaas.

Gedurende toetsing skakel AFTASKLOK 2 LAAG wat die aftasdata in houkring 1 inlees. Die uitset van houkring 1 word in houkring 2 geplaas as KLOK 2 na HOOG terugskakel.

Terwyl die werking van VSAO vlak-sensitief is as gevolg van die gebruik van twee aparte klokpulse vir houkringe 1 en 2, ontbreek vlak-sensitiwiteit by die aftasbaan tegniek aangesien net een klokpuls op 'n slag gebruik word (Hakim, 1989:106).

3.2.2.3 Aftas-stel logika (Scan-Set logic)

By die aftas-stel tegniek is die skuifregisters onafhanklik van alle stelselhoukringe deurdat dit nie deel van die databaan vorm nie. 'n Voorbeeld van aftas-stel word in figuur 3.7 aangetoon.

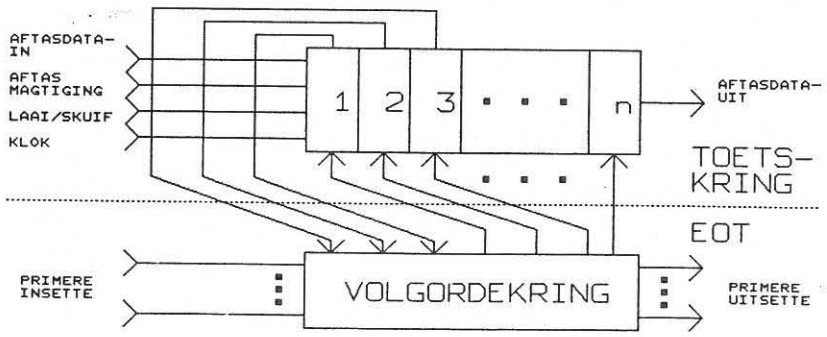


Fig. 3.7: Aftas-stel konfigurasie

Data word in die skuifregister ingelees deur middel van 'n puls op LAAI/SKUIF waarna dit met behulp van die klok by AFTASDATA-UIT van die skuifregister gelees word.

Vir die uitvoering van 'n toetsvektor word die verlangde data deur middel van 'n skuifproses in die skuifregister ingelees, en met behulp van die stelselklok na die betrokke wipkringe oorgedra.

'n Voordeel van hierdie tegniek is dat 'n aanduiding verkry word van die werking van die stelsel tydens normale werking, sonder om die stelsel te beïnvloed. 'n Voordeel van die tegniek bo VSAO en aftasbaan stelsels is die keuse wat uitgeoefen kan word oor watter houkringe gelees of gestel moet word.

3.2.2.4 Willekeurige-toegang-aftas (Random access scan)

Hierdie tegniek verskil van die voriges aangesien geen skuifregisters in die toetskring gebruik word nie. 'n Tegniek van adressering word gebruik vir die direkte beheer en waarneming van elke houkring. Twee tipes houkringe word benodig vir willekeurige-toegang-aftas toetsing en word in figure 3.8 en 3.9 aangetoon.

Figuur 3.8 toon 'n enkele houkring met 'n addisionele inset wat as AFTASDATA-IN (ADI) bekend

staan. Hie :d bemonster deur gebruik te maak van die AFTASDATAKLOK (ADK). Slegs indien adresse X en Y HOOG is sal hierdie klok die werking van die geselekteerde houkring beïnvloed. Met X=Y=1 kan data ook by AFTASDATA-UIT (ADU) gelees word.

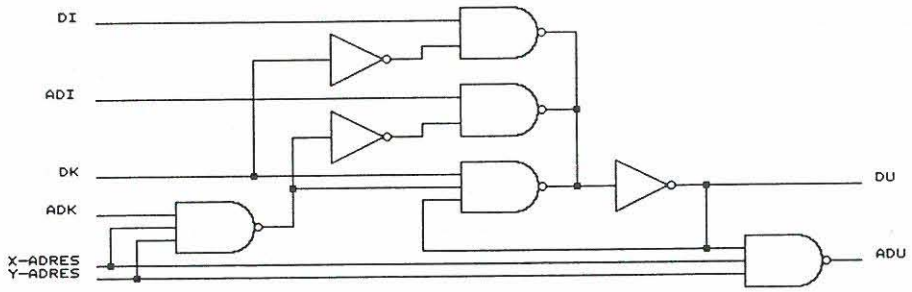


Fig. 3.8: Adresseerbare D-tipe houkring

Die houkring in figuur 3.9 is 'n stel-herstel tipe houkring wat sonder 'n aftasdataklok funksioneer. Deur middel van HERSTEL - wat gemeenskaplik aan ander houkringe is - word DU=0 gestel. 'n Houkring kan gestel word (DU=1) vir 'n toets deur die betrokke houkring te adresseer en dan 'n puls op die STEL-leiding toe te pas.

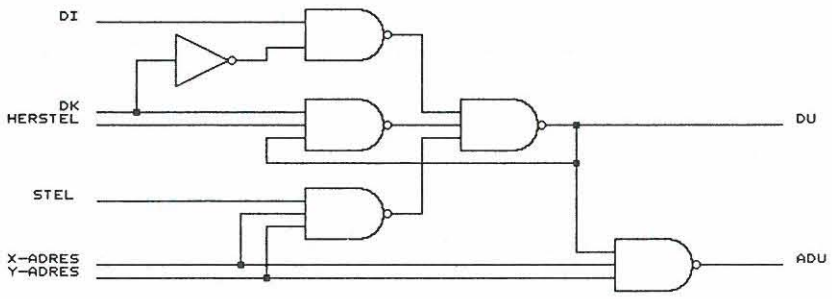


Fig. 3.9: Stel-herstel adresseerbare houkring

Figuur 3.10 toon die algemene konfigurasie van die willekeurige-toegang-aftas tegniek. Elke node wat

gemonitor [illegible] deur 'n unieke adres geïdentifiseer en kan sodoende van toetsdata voorsien word.

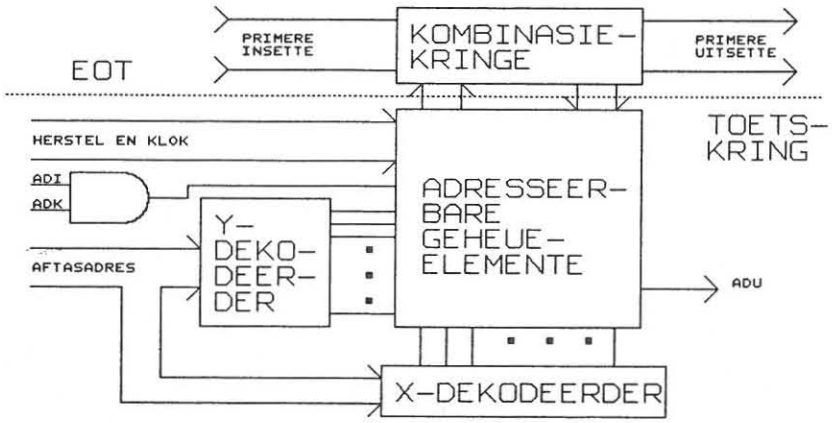


Fig. 3.10: Willekeurige-toegang-aftas tegniek

3.2.2.5 Onvoltooide-aftasbaan ontwerp

By die aftastegnieke wat tot dusver bespreek is, word gebruik gemaak van die uitbreiding van 'n kring in 'n kombinasie deel en 'n uitgebreide aftasbaan (toetskring). Dikwels is die uitbreiding van die kring onmoontlik as gevolg van die voorkoms van asinkrone volgordelogika. By die onvoltooide-aftasbaan tegniek word sommige geheue-elemente nie in die aftasbaan ingesluit nie.

Vir gebruik van die onvoltooide-aftasbaan ontwerp is 'n aantal reëls neergelê (Hakim, 1989:111).

i) Alle verbindings tussen die aftasbaan en die toetskring moet 'n uitset wees vanaf die aftasbaan en 'n inset na die toetskring.

ii) Die diepte van die volgorde gedeelte wat nie deel uitmaak van die aftasbaan nie, behoort tot drie beperk te word.

iii) Alle geheue-elemente wat maklik vanaf die primêre insette en -uitsette beheer en waargeneem kan word, moet nie in die aftasbaan ingesluit word nie.

iv) Indien asinkrone geheue-elemente, wat maklik beheer kan word, nie in die aftasbaan ingesluit kan word nie, moet sodanige elemente van die aftasbaan geïsoleer word.

3.2.3 Ingeboude-selftoetsing

Ingeboude-selftoetsing is 'n tegniek wat gebruik word om 'n kring te toets sonder die gebruik van enige eksterne toetsapparaat. Toetsvektore word in die kring opgewek, toegepas en die resultate in die kring gestoor.

Aangesien die toetskring deel vorm van die EOT, kan toetsing teen die normale spoed van die kring gedoen word. Voor gebruik van die toetskring moet die werking daarvan geverifieer word.

Ingeboude-logika-blok-waarnemer (ILBW) en sinjatuur-analise is ingeboude-selftoets tegnieke wat algemeen gebruik word.

3.2.3.1 Sinjatuuranalise (Signature Analysis)

Sinjatuuranalise is die opgewekking en kombinering van vektore deur gebruik te maak van 'n lineêre-terugvoer-skuifregister (LTSR). Figuur 3.11 toon so 'n konfigurasie.

Fig. 3.11: Algemene konfigurasie van sinjatuur-analise

'n LTSR word gebruik vir die opwekking van toetsvektore vir die EOT, sowel as die kombinering van die data wat by die uitset verkry word. Die uitset van die LTSR staan as die sinjatuur bekend. Deur die signatuur te vergelyk met die van 'n foutvrye kring, kan enige afwykings in die werking van die kring geïdentifiseer word. Die sinjatuur van 'n foutvrye kring word bepaal deur middel van simulasie.

Met elke klokpuls van die kring word data, vanaf nodes in die EOT, in 'n skuifregister ingelees. Die finale waarde in die register word enkodeer en die resultaat as heksadesimalesyfers aangedui. Elke sinjatuur is 'n getal wat die logika aktiwiteit van 'n spesifieke node gedurende 'n spesifieke tydinterval aantoon. Enige afwyking in die sinjatuur sal 'n aanduiding wees van 'n defek in die kring.

Swak diagnosering van foute word verkry in kringe met terugvoering tydens sinjatuuranalise (Turino, 1990:308). Die volgende voordele kan egter verwag word voortspruitend uit die gebruik daarvan:

i) Groot hoeveelhede toetse kan toegepas word teen die werkspoed van die kring.

ii) Groot hoeveelhede resultate kan gekombineer word in 'n sinjatuur.

iii) Vinnige bepaling van toetsvektore is gewoonlik moontlik.

3.2.3.2 Ingeboude-logika-blok-waarnemer (Build-In Logic Block Observer)

By 'n ingeboude-logika-blok-waarnemer (ILBW) word van die tegnieke van aftasbaan en VSAO geïntegreer met die sinjatuuranalise konsep deur die LTSR as 'n aftasregister, patroongenerator of sinjatuuranaliseerder te gebruik. 'n Kring wat gebruik maak van die ILBW tegniek word in figuur 3.12 aangetoon.

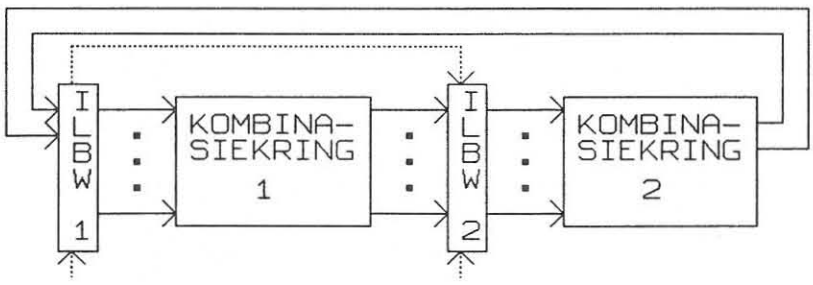


Fig. 3.12: ILBW toetsing

Deur middel van beheerleidings kan die modus waarin die ILBW funksioneer beheer word. Die ILBW by die

inset van kring 1 funksioneer as 'n patroongenerator terwyl ILBW 2 as 'n sinjatuurgenerator funksioneer. Die uitset van ILBW 1 is 'n willekeurige patroon van data en die uitset van kring 1 word in ILBW 2 gestoor.

Die modus van werking keer nou om en ILBW 2 funksioneer as 'n patroongenerator en ILBW 1 as 'n signatuurgenerator ter evaluering van kombinasie-kring 2. Die toetsing van beide kringe 1 en 2 word teen hoë snelhede uitgevoer. Diagnosering behels ontleding van die patrone in die skuifregisters.

Soos by sinjatuuranalise word die toetspatrone deur die kring self genereer. Indien die ILBW tegniek as 'n ingeboude toetstegniek gebruik word, moet foutsimulasie gedoen word om die effektiwiteit van die toets te bepaal.

3.3 Opsomming

Kringontwerp en kringtoetsing kan nie meer as twee aparte tereine in die vervaardiging van elektroniese borde gesien word nie. Om kostes so laag as moontlik te hou moet dié twee tereine van meet af saam beplan word (Spencer en Savir, 1985:287).

Om toetsbaarheid - en dus die nodige foutdekking - van 'n kring te verseker moet ontwerp-om-te-kan-toets tegnieke toegepas word. Tegnieke wat gebruik kan word wissel vanaf

eenvoudige Ad hoc tegnieke tot strukturele aftas tegnieke en ingeboude-selftoetsing.

UITEENSETTING VAN DIE AUTOMATIESE-TOETSSTELSEL

4.1 Inleiding

Die outomatiese-toetsstelsel bestaan uit 'n persoonlike-rekenaar, 'n koppelvlakkaart en bedryfsagteware. Toetse wat in die verlede met die hulp van verskillende toetsinstrumente gedoen is, word in die OTS deur sagteware verrig. Die OTS sal ten opsigte van hardeware en sagteware bespreek word.

4.2 Hardeware

Die hardeware van die outomatiese-toetsstelsel bestaan uit 'n rekenaar, koppelvlakkaart, en die hegstuk tussen die koppelvlakkaart en die EOT. Elkeen van die komponente word kortliks bespreek.

4.2.1 Rekenaar

Vir die gebruik as 'n outomatiese-toetsstelsel kan van enige IBM PC/XT/AT of IBM aanpasbare persoonlike-rekenaar gebruik gemaak word.

4.2.2 Koppelvlakkaart

'n Koppeling tussen die rekenaar en die EOT moet bestaan vir die oordrag van die toetsvektore vanaf die rekenaar na die EOT en vir die versameling van die resultate

vanaf die EOT. In baie gevalle maak vervaardigers van outomatiese-toetsstelsels gebruik van die IEEE-488 as standaard (Anoniem, 1990:309). Deur middel van hierdie koppeling word eksterne toetsinstrumente deur 'n rekenaar beheer. Vir die doel van hierdie projek word egter slegs van 'n rekenaar gebruik gemaak - alle toetse word dus deur 'n rekenaar beheer en uitgevoer.

Aangesien digitale borde getoets word - die data bestaan dus hoofsaaklik uit 0e en 1e - kan van die PC-14A koppelvlakkaart gebruik gemaak word. Die PC-14A is 'n programmeerbare inset/uitset (I/U) bord aanpasbaar met die IBM PC/XT/AT persoonlike rekenaar (Eagle Electronic Catalogue, 1991:43). Die PC-14A kaart kan in enige beskikbare uitbreidingsgleuf in die rekenaar geplaas word.

Die PC-14A kaart voorsien 48 programmeerbare I/U leidings asook drie tellers/tydmeters, wat gebruik kan word vir die telling van gebeurlikhede of die bepaling van die lengte van 'n gebeurtenis.

Die PC-14A kaart word vervaardig rondom die gebruik van twee 8255 programmeerbare-rand-koppelvlak (PRK) integreerde stroombane. 'n 8253 Integreerde stroombaan word gebruik vir die teller/tydmeter funksies.

4.2.2.1 Poortadresse

Die PC-14A maak van twee groepe adresspasies in die I/U bus gebruik, naamlik 1B0-1BF en 1F0-1FF. Met skakelaar SW4 (op die PC-14A) aangeskakel word adresspasie 1B0-1BF gebruik terwyl adresspasie 1F0-1FF gebruik word indien skakelaar SW5 aangeskakel is. Die gebruik van die adresspasies word in tabel 4.1 uiteengesit. Slegs SW4 word aangetoon aangesien dié betrokke adresspasie gebruik is.

Tabel 4.1 Adres uiteensetting van PC-14A kaart (SW4 aangeskakel)

Poortadres	Funksie
1B0 (432)	8255 Nommer 1 Poort A
1B1 (433)	8255 Nommer 1 Poort B
1B2 (434)	8255 Nommer 1 Poort C
1B3 (435)	8255 Nommer 1 Beheerpoort
1B4 (436)	8255 Nommer 2 Poort A
1B5 (437)	8255 Nommer 2 Poort B
1B6 (438)	8255 Nommer 2 Poort C
1B7 (439)	8255 Nommer 2 Beheerpoort
1B8 (440)	8253 Teller/Tydmeter 0
1B9 (441)	8253 Teller/Tydmeter 1
1BA (442)	8253 Teller/Tydmeter 2
1BB (443)	8253 Teller/Tydmeter Beheer

4.2.2.2 Eksterne konneksies van die PC-14A

Koppeling van die PC-14A kaart is deur middel van twee 40-pen dubbelry penkonnekteerders en word in tabel 4.2 aangetoon.

Tabel 4.2 Eksterne konneksies van PC-14A

KANAAL 1 (POORT 1)				KANAAL 2 (POORT 2)			
PEN	**en FUNKSIE**			**PEN**	**en FUNKSIE**		
1	GND	21	PC6	1	GND	21	PC7
2	GND	22	PC7	2	GND	22	PC6
3	G/V	23	PC4	3	G/V	23	PC5
4	PA3	24	PC5	4	G/V	24	PC4
5	PA1	25	PC1	5	G/V	25	PC0
6	PA2	26	PC0	6	G/V	26	PC1
7	KLKIN_0	27	PB7	7	G/V	27	PC2
8	PA0	28	PC2	8	G/V	28	PB7
9	HEK_0	29	PB6	9	G/V	29	PC3
10	KLKUIT_0	30	PC3	10	G/V	30	PB6
11	KLKUIT_2	31	PB5	11	G/V	31	PB0
12	KLKIN_2	32	PB0	12	G/V	32	PB5
13	KLKIN_1	33	PB4	13	PA0	33	PB1
14	HEK_2	34	PB1	14	PA1	34	PB4
15	KLKUIT_1	35	PB3	15	PA2	35	PB2
16	HEK_1	36	PB2	16	PA3	36	PB3
17	PA5	37	+5V	17	PA4	37	+5V
18	PA4	38	-5V	18	PA5	38	-5V
19	PA7	39	+12V	19	PA6	39	+12V
20	PA6	40	-12V	20	PA7	40	-12V

G/V = geen verbinding

4.2.3 Hegstuk tussen koppelvlakkaart en die EOT

Ten einde toetsvektore toe te pas op die EOT, en om die resultate terug te lees, word 'n hegstuk (fisiese verbinding) tussen die PC-14A en die EOT benodig. Die toetsstelsel maak voorsiening vir die toetsing van borde wat van 'n randkonnekteerder voorsien is. Die PC-14A kaart en die EOT word deur middel van 'n koppelvlakkabel verbind.

4.3 Sagteware

4.3.1 Programmeringsfilosofie

Die outomatiese-toetsstelsel het die voordeel van geen interaksie deur die operateur tydens toetsing nie aangesien alle besluitneming deur die rekenaar gedoen word. Die toetsprogram voer die toetsvektore uit, lees die resultate terug vanaf die EOT en vergelyk dit met die foutvrye uitsette wat bekend is. Indien 'n foutiewe uitset verkry word, word addisionele toetse uitgevoer, totdat die foutiewe node geïdentifiseer kan word.

Aangesien 'n verskeidenheid van funksies in die sagteware van die OTS ingebou is, is die program rondom 'n menu geskryf wat die gebruik van die stelsel baie vereenvoudig. Die uitleg van die menu met alle sub-menus word in figuur 4.1 aangetoon. Turbo Pascal is gebruik vir die skryf van die sagteware vir die OTS.

Die menu word in drie kategorieë verdeel soos aangetoon in figuur 4.1. Sub-menu een word gebruik vir funksies rondom die uitvoering van 'n toets, terwyl sub-menus twee en drie onderskeidelik vir die skep van 'n nuwe lêer (of databasis) en die verandering van 'n bestaande lêer gebruik word. 'n Lêer word geskep vir elke bord wat deur die OTS getoets moet word.

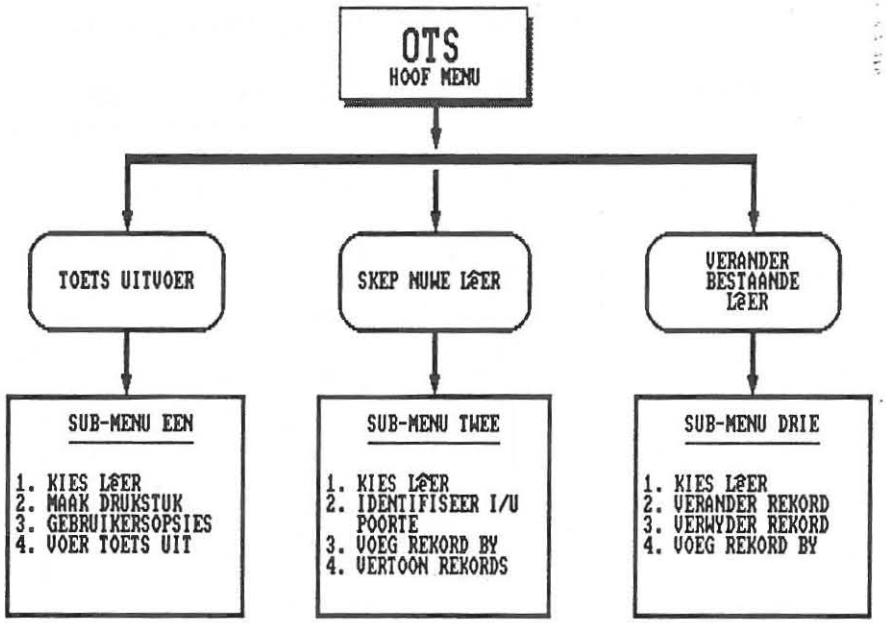


Fig.4.1: Menu uitleg van die outomatiese-toetsstelsel

Elke sub-menu word vervolgens bespreek. Om die verduideliking te vereenvoudig word sub-menu twee eers bespreek gevolg deur sub-menus een en drie.

4.3.2.1 Sub-menu twee (Skep nuwe lêer)

'n Lêer bestaan uit rekords wat elk uit vyf velde bestaan. Hierdie velde word gedefinieer as

toetsnommer, vektor, resultaat (of foutvrye-uitset) en die slaag en faal instruksievelde. Ter illustrering van die opstelling van toetsrekords word daar van 'n uittreksel uit 'n toetsstel gebruik gemaak soos aangetoon in Tabel 4.3. Velde wat nie in die voorbeeld gebruik word nie bevat geen inligting. Alle inligting in die vektor- en resultaatvelde word as desimale getalle aangetoon.

Tabel 4.3 Uittreksel van toetsstel

Rekord-nommer	Toets-nommer	Vektor	Resultaat	Slaag	Faal
0	ADDR	131	128		
1	CTRL	436	2		
2	O_PRT	432			
3	I_PRT	433	7		
4	CLK	432	254		
5	CLK	1	246		
6	1t3	3	6	1t7	2t7
7	1t7	7	0	1t10	3t10
8	1t12	12	1	1t15	5t15
9	1t15	15	7	f0	f1
10	END				
11	f0	1			
12	FFREE				
13	f1	2			
14	data0/0			u1-1/0	u1-9/0
15	u3-13/0				
16	END				
17	END				

Afgesien van die inligting soos deur die onderskeie veldname aangedui, kan in Tabel 4.3 gesien word dat velde in sommige rekords ook vir ander inligting gebruik word. Van hierdie inligting is ondermeer poortinligting, beheerpoortinligting, foutinligting en klokinformasie. Die toetsnommerveld word gebruik om vir die rekenaar 'n aanduiding te gee van watter inligting in die betrokke rekord se velde verwag kan word.

Die toetsnommer-, slaag- en faalvelde word elk as 'n string van agt karakters gedefinieer terwyl die vektor- en resultaatvelde as heelgetal velde gedefinieer word. Vervolgens word elk van die funksies van sub-menu twee bespreek:

i) Kies lêer

Soos by die ander sub-menus word die eerste opsie gebruik om 'n bestaande lêer te kies of 'n nuwe te skep. 'n Lêer kan gekies word deur die lêernaam - tesame met die pad (path) waar die spesifieke lêer gevind kan word - in te sleutel. Die "*" karakter kan gebruik word indien 'n lys van lêers verlang word. Vanuit die lys kan 'n spesifieke lêer gekies word.

Die lêer wat gekies word, word as die werklêer aangedui en alle verdere instruksies sal hierop van toepassing wees. 'n Nuwe lêer kan geskep word deur die naam van die nuwe lêer in te

sleutel tesame met die lêernaamuitbreiding ".TSD".

ii) Identifisering van inset/uitset (I/U) poorte

Elke tipe bord wat getoets word het unieke vereistes ten opsigte van insette en uitsette. Met die identifisering van die I/U poorte word elke poort op die twee 8255's as 'n inset of uitset gekies deur middel van 'n opsieblok op die skerm. Vanaf hierdie inligting word die beheerwoord van elke 8255 PRK outomaties bereken en in die toetsstel bygewerk.

Die beheerwoorde word as die eerste rekord (rekordnommer 0) van elke lêer gestoor en word deur middel van "ADDR" in die toetsnommerveld aangedui. Die beheerwoorde van kanaal een en twee word onderskeidelik in die vektor- en resultaatvelde aangedui. In Tabel 4.3 is die beheerwoorde van kanaal een en twee - vir die betrokke bord - onderskeidelik 131 en 128.

iii) Voeg rekord by

Indien hierdie opsie uitgevoer word kan 'n verskeidenheid inligting as rekords bygevoeg word naamlik toetsrekords, inset- en uitset-poorte, beheerpoorte, klokinligting en fout-informasie. Die inligting word kortliks bespreek.

■ Voeg beheerpoort by (CTRL)

Met die toetsing van 'n bord word konstante waardes (0e of 1e) deur die verloop van die toets op sommige insette van die kring benodig. 'n voorbeeld van 'n konstante waarde is die LAAG wat gedurende toetsing aan die toetsleiding van figuur 3.1 verbind word. Hierdie waardes kan verskaf word deur 'n sogenaamde beheerpoort.

'n Beheerpoort word in die toetsnommerveld aangedui as "CTRL" en moet nie verwar word met die beheerwoord van die 8255 nie (aangedui as "ADDR"). Die beheerpoort (volgens tabel 4.1) en die beheerwoord word onderskeidelik in die vektor- en die resultaatvelde aangetoon. Rekordnommer 1 in Tabel 4.3 dui 'n beheerwoord van 2 op poort 436 (1B4H) aan.

■ Voeg poort by (I_PRT of O_PRT)

Afhangend van die aantal insette en uitsette wat vir 'n spesifike kring benodig word, word inset- en uitsetpoorte van die OTS as sulks geprogrammeer. Die volgorde waarin hierdie poorte verskaf word sal die volgorde wees waarin die vektore na die EOT uitgeskryf word en die resultate teruggelees word. Enige kombinasie van inset- en uitsetpoorte kan gebruik word terwyl die aantal poorte slegs beperk word deur die maksimum aantal poorte wat op die twee 8255 PRKs beskikbaar is - naamlik 6 x 8 bis poorte.

Die toetsnommerveld sal die woorde "O_PRT" of "I_PRT" bevat afhangend daarvan of die poort waarna verwys word 'n uitset- of insetpoort moet wees. Die vektorveld word gebruik om 'n poort - soos verskaf in tabel 4.1 - te identifiseer. Indien 'n insetpoort verlang word moet die maskeergreep vir die maskering van die resultaat in die resultaatveld verskaf word.

■ Klok (CLK)

Klokpulse word hoofsaaklik gebruik indien volgordekringe getoets moet word en word in die toetsnommerveld as "CLK" aangedui. Rekordnommers 4 en 5 in Tabel 4.3 is 'n voorbeeld van 'n klokpuls. Soos gesien kan word, word twee rekords benodig vir klokinformasie. In die eerste rekord word die vektorveld gebruik om aan te dui watter poort (volgens tabel 4.1) gebruik moet word vir die klok. Die resultaatveld word gebruik om die begin pulswaarde (logika 0 of 1) aan te toon. In die tweede rekord dui die vektorveld die hoeveelheid pulse aan wat uitgevoer moet word terwyl die resultaatveld die end pulswaarde aandui.

Deur na die voorbeeld te kyk kan gesien word dat die begin- en end pulswaardes onderskeidelik 254 en 246 is. Indien hierdie waardes in binêr geskryf word, kan die resultaat beter gesien word:

254 1111 1110B

246 1111 0110B

Dit is duidelik dat bis drie verander vanaf 'n HOOG na 'n LAAG. Die betrokke bis verander dus vanaf 'n LAAG na 'n HOOG en keer daarna terug na 'n LAAG. Deur die korrekte keuses van begin- en end pulswaardes kan 'n enkele bis op enige uitsetpoort as 'n klok gebruik word (ongebruikte bisse bly konstant). Aktief laag sowel as aktief hoë pulse kan genereer word.

■ Voeg toetsrekord by

Hierdie inligting word opgestel soos deur die benamings van die onderskeie velde aangetoon. In Tabel 4.3 is rekordnommers 6 tot 9 voorbeelde van toetsrekords.

Aangesien toetsvektore nie noodwendig op mekaar volg nie word die toetsnommer in die toetsnommerveld gebruik as 'n aanduiding vir die rekenaar watter toetsvektor uitgevoer moet word. Die formaat wat gebruik word is soos aangetoon in Tabel 4.3. Dit kan geskryf word as xxxtxxx, waar die nommer voor die t gebruik word om onderskeid tussen groepe toetsvektore te tref. Elke been van 'n foutdiagram (figuur 2.9) vorm 'n groep toetsvektore. Die syfer na die t word dieselfde as die betrokke vektor gemaak. Die inligting in die slaag- en faalvelde - volgens die uitslag van die betrokke toetsvektore - word

gebruik om die stelsel te lei na 'n volgende toets.

Die vektor- en resultaatveld toon onderskeidelik die toetsvektor en die verwagte foutvrye-uitset van die betrokke toetsvektor in desimale waardes aan. Deur die werklike uitset van die kring met die verwagte foutvrye-uitset te vergelyk kan bepaal word of die toetsvektor geslaag word al dan nie. Die einde van die toetse word deur middel van END in rekordnommer 10 aangetoon.

■ Foutinligting

Nadat foutdiagnosering gedoen is moet die foutinligting aan die gebruiker oorgedra word. Ter illustrering hiervan word weer na Tabel 4.3 verwys. Veronderstel die toets is uitgevoer tot by die vlak waar die volgende toetsnommer gelyk aan f1 is. Vir 'n verduideliking van die uitvoering van 'n toets word na 4.3.2.2 (iii) (voer toets uit) verwys. F1 is 'n foutnommer (soos behandel in 2.8) wat die rekenaar gebruik om aan te dui dat 'n sekere node by 'n voorafbepaalde, foutiewe vlak is. Addisionele inligting, wat die foutiewe node en dies meer aantoon, moet nou aan die gebruiker verskaf word.

Tydens toetsing beweeg die rekenaar deur die toetsstel (nadat f1 byvoorbeeld in die slaag- of

faalvelo voorgekom net) totdat dit f1 in die toetsnommerveld aantref (rekordnommer 13 in Tabel 4.3). Aangesien 'n toetsuitslag nie altyd tot 'n enkele node beperk kan word nie, word daar soms van meer as een rekord gebruik gemaak om die foutinligting van 'n betrokke fout te stoor. Die aantal rekords wat benodig word, word in die vektorveld aangetoon (2 rekords in die voorbeeld). Die foutinligting word in die toetsnommer-, slaag- en faalvelde gestoor soos aangetoon in rekordnommers 14 en 15 in Tabel 4.3.

Indien die uitslag van die toets f0 is dui dit daarop dat die betrokke kring (of gedeelte van 'n kring) foutvry is. Dit word aan die gebruiker vertoon as FFREE in rekordnommer 12.

Die einde van die lêer word deur middel van 'n dubbele END aangedui soos aangetoon in rekordnommers 16 en 17.

iv) Vertoon rekords

Met hierdie opsie word die inhoud van die werklêer op die skerm vertoon. Rekordnommers soos aangedui in Tabel 4.3 word outomaties deur die OTS bygevoeg (word nie in die lêer gestoor nie) wat dan gebruik word vir die verandering, verwydering of byvoeging van rekords. Bladwisseling ("Page Up", "Page Down", "Home" en

"End") kan gebruik word om die inligting in die lêer na te gaan.

4.3.2.2 Sub-menu een (Toets uitvoer)

i) Maak drukstuk

'n Drukstuk kan van die werklêer asook van die resultaatlêer - wat outomaties deur die OTS geskep word nadat 'n toets uitgevoer is - gemaak word. Die resultaatlêer (lêernaamuitbreiding ".TSR") is 'n tydelike lêer wat die verloop van 'n afgehandelde toets aandui.

ii) Gebruikersopsies

Die gebruikersopsies verskaf 'n aantal keuses wat die gebruiker kan uitoefen wat die verloop van die toets sal bepaal. Dié opsies kan deur middel van 'n opsieblok gekies word en sluit ondermeer die volgende in:

- Isoleer fout

  Met hierdie opsie word daar gekies of die fout geïsoleer moet word tot die kleinste moontlike vervangbare komponent en of die bord slegs as reg of foutief aangedui moet word. Laasgenoemde toets word vinniger uitgevoer.

- Enkelstap deur toets

  Hierdie opsie kan baie handig gebruik word

wanneer diagnosering met die hand gedoen moet word nadat toetsing deur die rekenaar afgehandel is. Deur stap-vir-stap deur elke toets te gaan kan vir besondere kondisies op sekere nodes getoets word. Hierdie metode kan baie met verdere diagnosering help in gevalle waar die rekenaar 'n fout nie tot 'n enkele node kan diagnoseer nie.

- Vertoon alle stappe

  Deur gebruik van hierdie opsie kan die gebruiker bepaal of elke toets en resultaat op die skerm vertoon moet word al dan nie. Indien verlang sal slegs 'n "*" vertoon word na die uitvoering van elke toetsvektor as aanduiding dat die rekenaar besig is met die uitvoering van 'n toets.

- Resultaatlêer uitwis

  Na die voltooiing van elke volledige toets sal die OTS 'n resultaatlêer genereer. Indien verlang kan die gebruiker sodanige lêer uitwis.

iii) Voer toets uit

Die uitvoering van 'n toets word deur middel van 'n vloeidiagram in figuur 4.2 voorgestel. Soos genoem in 3.2.1.1 kan 'n kring opgedeel word in kleiner kringe om toetsing te vereenvoudig. Elke kring wat so gevorm word, word afsonderlik

getoets gedoen aangesien elke sub-kring unieke inset- en uitsetpoorte asook beheerpoorte het.

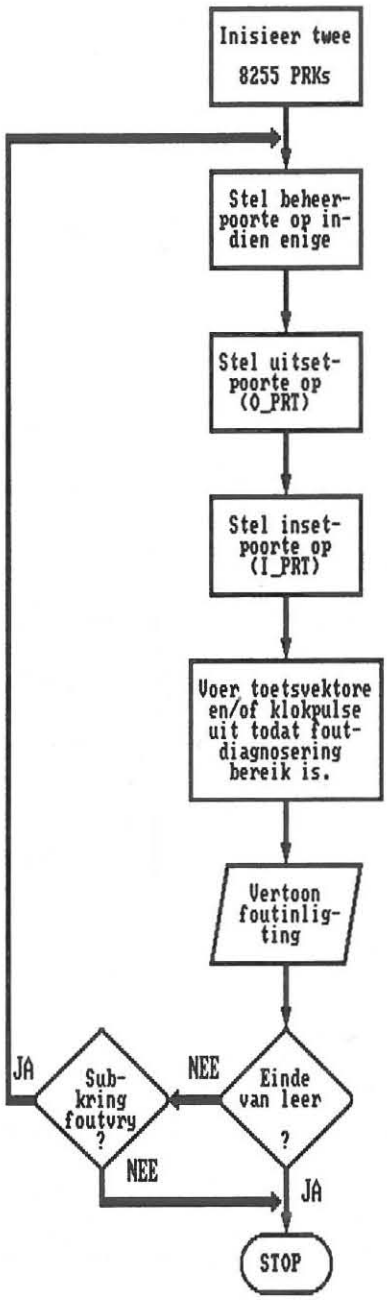


Fig. 4.2: Vloeidiagram van die uitvoering van 'n toetsstel.

Inisieering van die twee 8255 PRKs word gedoen deur middel van die beheerwoorde wat in die vektor- en resultaatvelde van rekord zero (sien

Tabel 4.3) in elke lêer voorkom. Indien enige beheerpoorte (CTRL) benodig word, word dit as sodanig opgestel. Alle uitsetpoorte, gevolg deur die insetpoorte, word vervolgens opgestel. Die volgorde waarin die poorte voorkom is baie belangrik, aangesien dit die volgorde is waarin toetsvektore uitgeskryf word en resultate teruggelees word.

Na inisieering afgehandel is word toetsvektore en/of klokpulse na die EOT gestuur en die resultate gelees todat foutdiagnosering bereik word. Die uitvoer van die toets kan verduidelik word deur na Tabel 4.3 (rekordnommer 6) te verwys. Die rekenaar stuur vektor 3 na die PC-14A koppelvlakkaart wat dit aan die EOT koppel. Die resultate word vanaf die EOT gelees en vergelyk met die foutvrye-uitset van die toets.

Afhangend van of die toetsvektor slaag of faal word addisionele toetse uitgevoer todat 'n defek in die EOT geïdentifiseer word, of die foutvrye werking van die kring (of sub-kring) bevestig word. Die inligting in die slaag- en faalvelde lei die stelsel na die volgende toets wat uitgevoer moet word.

Indien foutdiagnosering bereik is word die inligting aan die gebruiker gegee. Indien die einde van die lêer bereik is (END in

rekordnommers 16 en 17) word die toets gestop. Indien dit nie die einde van die lêer is nie en die kring is foutvry - wat daarop dui dat nog sub-kringe getoets moet word - word die hele proses herhaal vir sodanige kringe. Die toets en foutinligting word geskei deur 'n enkele "END" in die toetsnommerveld te plaas (sien rekord-nommer 10).

4.3.2.3 Sub-menu drie (Verander bestaande lêer)

Met die hulp van hierdie opsies kan bestaande rekords verander of verwyder word. Addisionele rekords kan ook op enige plek in 'n lêer bygevoeg word. Die rekordnommers wat vertoon word indien die lêer op die skerm vertoon word, word gebruik om rekords te verander, te verwyder of by te voeg.

Slegs een rekord kan met elke uitvoering van die betrokke opsie verander of bygevoeg word. Daarteenoor kan enige aantal rekords - soos deur die gebruiker verskaf word - met 'n enkele instruksie verwyder word. Enige van die rekords soos in 4.3.2.1 (iii) bespreek is kan deur middel van die "voeg rekord by" opsie in die lêer gevoeg word. Dié opsie verskil van die van sub-menu twee deurdat 'n rekord op enige plek in die lêer bygevoeg kan word. Met sub-menu twee word die rekords telkens aan die einde van die lêer geplaas.

4.3.3 Toetsing van sagteware

Twee tipes probleme wat algemeen in sagteware voorkom is sintaksfoute en logikafoute. Sintaksfoute lewer nie 'n groot probleem nie aangesien die vertaler dié tipe foute uitwys indien die bronkode vertaal word. Logikafoute veroorsaak egter meer probleme aangesien dié tipe fout nie deur die vertaler geïdentifiseer sal word nie.

Toetsing van sagteware word hoofsaaklik onder funksionele- en strukturele toetsing geklassifiseer. Die grootste nadeel van funksionele toetsing is dat geen metode bestaan om te verseker dat toetsing voltooi is nie. Strukturele toetsing word gebruik om hierdie tekortkoming te oorbrug (Herington, Nichols en Lipp, 1987:13).

'n Metode wat gevolg kan word om logikafoute te identifiseer is deur gebruik te maak van 'n volledige toets. Met 'n volledige toets word daar van enkelstapping deur elke moontlike programbaan gebruik gemaak. As gevolg van die groot hoeveelheid besluit-neming wat egter in hierdie program voorkom is hierdie metode nie prakties uitvoerbaar nie.

Die metode wat gebruik is vir die toetsing van die sagteware - om so logikafoute te elimineer - is deur elke prosedure en funksie volledig te toets nadat dit voltooi is. Aangesien elke prosedure of funksie kort is, is dit moontlik om 'n volledige toets op elk uit te

voer. Nadat prosedures en funksies saamgevoeg is, is die program as 'n geheel getoets. Daar is gepoog om sover as moontlik alle instruksies in die program uit te voer. Enige prosedures of funksies wat na samevoeging oënskynlik nie korrek wou funksioneer nie, is volledig getoets deur gebruik te maak van enkelstapping van die ontfouter (debugger).

4.4 Opsomming

Die hoofkomponente in die OTS is die rekenaar met die gepaardgaande sagteware. Vir die oordrag van die binêre data vanaf die rekenaar na die EOT, word van 'n PC-14A koppelvlakkaart gebruik gemaak. 'n Randkonnekteerder word gebruik as hegstuk tussen die OTS en EOT.

Die sagteware van die OTS is rondom 'n menu geskryf. Die skepping van 'n nuwe lêer, die verandering van 'n bestaande lêer en die uitvoering van 'n toets word alles vanaf die menu uitgevoer.

Ter evaluering van die OTS is 'n aantal borde gebou en toetse op die kringe uitgevoer.

Evaluering van die toetsstelsel

5.1 Inleiding

Ter evaluering van die outomatiese-toetsstelsel is twee kringe gebou en 'n toetsstel vir elk bepaal. Hierdie toetsstelle is elk as 'n lêer geskep en op die kringe uitgeoefen. Ter evaluering van die effektiwiteit van die toetsstelle is al die foute - soos in die betrokke foutlys uiteengesit - op die kring gesimuleer.

Die twee kringe bestaan onderskeidelik uit kleinskaalse-integrasie (KSI) logika en meduimskaalse-integrasie (MSI) logika. Elk word vervolgens individueel bespreek.

5.2 KSI logika

Met die eerste kring is uitsluitlik van kombinasielogika gebruik gemaak. Hierdie kring bestaan dus slegs uit EN-, OF-, NEN-, NOF- en eksklusiewe OF-hekke en omkeerders.

5.2.1 Ontwerp van die kring

'n Eenrigting datakommunikasiestelsel is ontwerp wat gebruik maak van die Hammingkode tegniek vir data foutopsporing en herstelling. Die basiese uitleg van die Hammingkode tegniek word blokdiagrammaties in figuur 5.1(a) aangetoon.

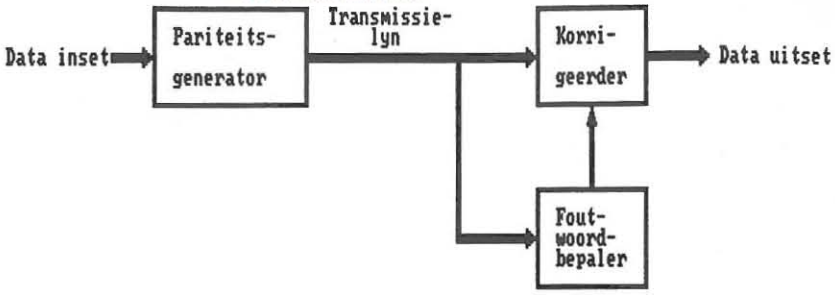


Fig.5.1(a): Hammingkode tegniek gebruik in 'n datakommunikasiestelsel.

Die sender en ontvanger bestaan onderskeidelik uit 'n Hammingkode generator en -toetser. Hierdie kring is uitgebrei - soos in figuur 5.1(b) aangetoon word - om toetsing van die kring te vereenvoudig.

5.2.2 Bepaling van die toetsvektore volgens segment.

Ontwerp-om-te-kan-toets tegnieke is gebruik om die bepaling van toetsvektore te vereenvoudig. In figuur 5.1(b) word addisionele insette, uitsette en beheerleidings aangetoon wat gebruik word vir die toetsing van die kring. Die kring onder bespreking is in drie dele opgedeel, naamlik die generator, foutwoord-bepaler en korrigeerder. Toetsvektore is vir elk van hierdie dele bepaal. Addisionele komponente is bygevoeg om hierdie opdeling moontlik te maak en om terselfdertyd addisionele insette en uitsette aan die randkonnekteer-der te verskaf.

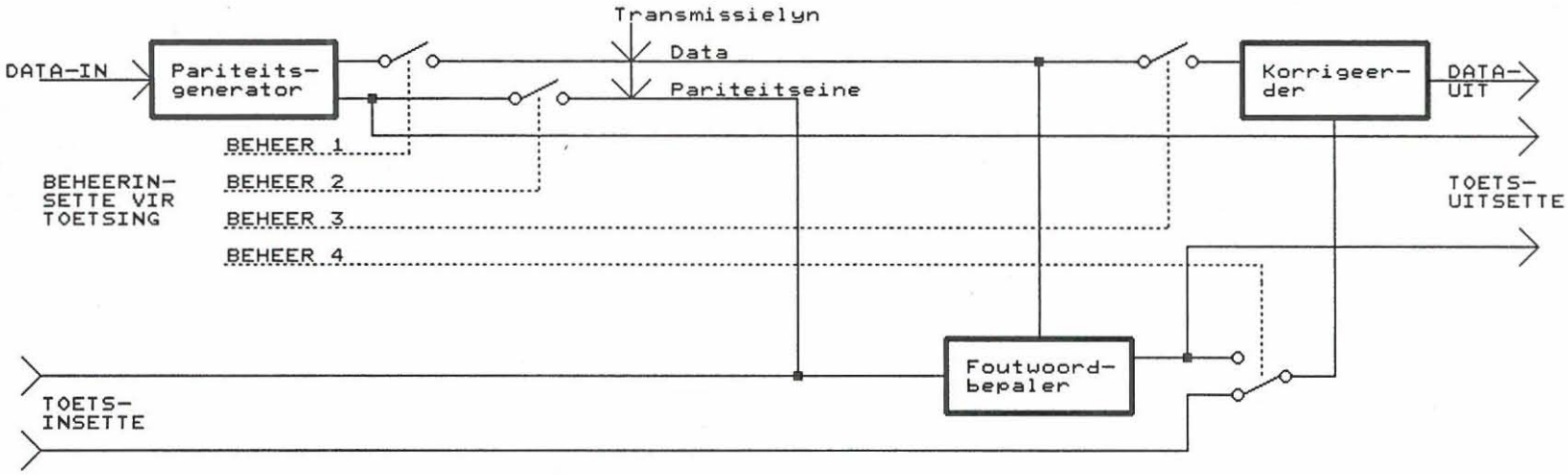


Fig. 5.1(b): Uitbreiding van die datakommunikasiestelsel om toetsing te vereenvoudig

Figuur 5.2 toon die generator met die addisionele komponente wat bygevoeg is vir die opdeling van die kring vir toetsdoeleindes. EN-hekke U3A tot U3D en drietoestandbuffers U4A tot U4C is addisionele komponente wat gebruik word om onderskeidelik die dataleidings (DATA 0 tot DATA 3) en die Hammingkode bisse (C1, C2 en C4) van die res van die kring te isoleer. Die drietoestandbuffers word gebruik aangesien insette vanaf die OTS aan die uitsette van hierdie komponente gekoppel word, om as insette na die foutwoordbepaler te dien.

Aanvanklik is bepaal hoeveel insette en uitsette vanaf die OTS na die kring benodig word. Vanaf figuur 5.2 blyk dat ses insette en drie uitsette benodig word. Beheerleidings 1 en 2 is gebruik vir die afskakeling van die EN-hekke en die drietoestandbuffers. Soos aangedui word hierdie leidings onderskeidelik aan bisse zero en een van poort 1B4H (436) verbind. In die lêer is poort 1B4H dus 'n beheerpoort (CTRL) gemaak terwyl die beheerwoord 02H was - sodat bis0 = 0 en bis1 = 1 is.

Bisse 0 tot 3 van poort 1B0H (432) word gebruik vir DATA 0 tot DATA 3 terwyl bisse 0 tot 2 van poort 1B1H (433) gebruik is vir uitsette 1 tot 3. Twee rekords moes in die lêer geskep word om hierdie poort konfigurasie aan te dui. Met die "O_PRT" word

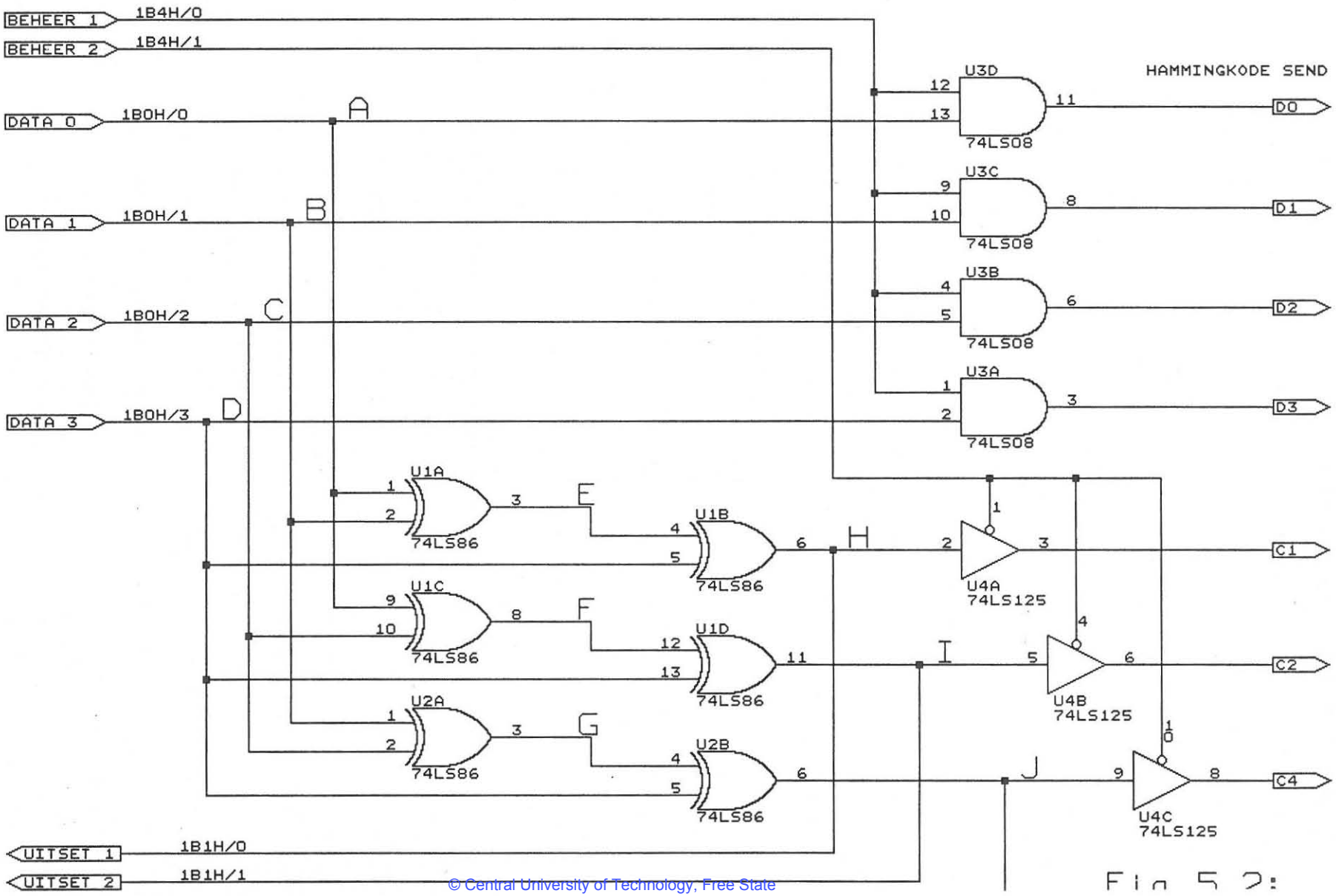


Fig 5.2:

slegs die (432) aangedui terwyl die "I_PRT" 1B1H (433) ook 'n maskeergreep van 7 (die drie lae bisse word gebruik) in die resultaatveld bevat. Die maskeergreep word benodig om ongewensde insette wat op die oorblywende bisse ingelees word te verwyder.

i) Bepaling van die foutlys

Die vasgeklem-by-waarde model is gebruik vir die bepaling van die foutlys. Met hierdie model kan elke node in die kring vasgeklem wees by 0 of 1. Om al die nodes maklik uit te ken word dit genommer vanaf A tot J soos aangedui in figuur 5.2. Die foutlys sal dus soos volg daar uitsien:

A/0, A/1, B/0, B/1, C/0, C/1, D/0, D/1, E/0, E/1, F/0, F/1, G/0, G/1, H/0, H/1, I/0, I/1, J/0, J/1.

ii) Bepaling van toetsvektore

Soos genoem in afdeling 2.5.3 kan 'n sensitiewe-pad vir enige hek - uitsluitend die eksklusiewe OF-hek - geskep word. Aangesien die generator slegs uit eksklusiewe OF-hekke bestaan veroorsaak dit probleme wanneer toetsvektore bepaal moet word. By 'n EN-hek en OF-hek kan byvoorbeeld onderskeidelik van 'n 1 en 0 gebruik gemaak word om die betrokke hekke te aktiveer terwyl albei hierdie kondisies geldig is by die eksklusiewe OF-hek.

Om hierdie probleem te oorkom - en aangesien die generator slegs vier insette het - kan daar van 'n waarheidstabel gebruik gemaak word vir die bepaling van die toetsvektore. Die toetsnommer, vektor en die foutvrye-uitset (op uitsette H tot J) van elke vektor word in tabel 5.1 aangetoon.

Tabel 5.1: Waarheidstabel vir die bepaling van toetsvektore vir generator.

Toets-nommer	Vektor	Foutvrye-uitset
	DCBA	JIH
t0	0000	000 (0)
t1	0001	011 (3)
t2	0010	101 (5)
t3	0011	110 (6)
t4	0100	110 (6)
t5	0101	101 (5)
t6	0110	011 (3)
t7	0111	000 (0)
t8	1000	111 (7)
t9	1001	100 (4)
t10	1010	010 (2)
t11	1011	001 (1)
t12	1100	001 (1)
t13	1101	010 (2)
t14	1110	100 (4)
t15	1111	111 (7)

Uit die foutlys en tabel 5.1 kan die volgende toetsvektore bepaal word:

Tabel 5.2: tsvektore vir foute op nodes A tot D van

Toets	Kondisie	Moontlike toetsvektore
A/0	A=1	t1, t3, t5, t7, t9, t11, t13, t15
A/1	A=0	t0, t2, t4, t6, t8, t10, t12, t14
B/0	B=1	t2, t3, t6, t7, t10, t11, t14, t15
B/1	B=0	t0, t1, t4, t5, t8, t9, t12, t13
C/0	C=1	t4, t5, t6, t7, t12, t13, t14, t15
C/1	C=0	t0, t1, t2, t3, t8, t9, t10, t11
D/0	D=1	t8, t9, t10, t11, t12, t13, t14, t15
D/1	D=0	t0, t1, t2, t3, t4, t5, t6, t7

Die bepaling van toetsvektore vir die interne nodes E tot G vereis meer berekeninge en word vervolgens bepaal. Die moontlike toetsvektore word in tabel 5.3 in die formaat DCBA aangedui waar X om't-ewe kondisies is (0 of 1).

Tabel 5.3: Moontlike toetsvektore vir foute op nodes E tot G van generator.

Toets	Kondisie	Moontlike toetsvektore
E/0	E=1	0X10, 1X10, 0X01, 1X01
E/1	E=0	0X00, 1X00, 0X11, 1X11
F/0	F=1	01X0, 11X0, 00X1, 10X1
F/1	F=0	00X0, 10X0, 01X1, 11X1
G/0	G=1	010X, 110X, 001X, 101X
G/1	G=0	000X, 100X, 011X, 111X

Vanaf tabel 5.3 kan 'n volledige lys van toetse bepaal word deur die X'e met 0 en 1 te vervang. Hierdie lys word in tabel 5.4 aangetoon.

Tabel 5.4: Volledige lys van moontlike toetsvektore vir foute op nodes E tot G van generator.

Toets	Kondisie	Moontlike toetsvektore
E/0	E=1	t1, t2, t5, t6, t9, t10, t13, t14
E/1	E=0	t0, t3, t4, t7, t8, t11, t12, t15
F/0	F=1	t1, t3, t4, t6, t9, t11, t12, t14
F/1	F=0	t0, t2, t5, t7, t8, t10, t13, t15
G/0	G=1	t2, t3, t4, t5, t10, t11, t12, t13
G/1	G=0	t0, t1, t6, t7, t8, t9, t14, t15

Deur van dieselfde redenasie as die van die toetsvektore van nodes A tot D gebruik te maak, is die toetsvektore vir die moontlike foute op nodes H, I en J ook vanaf die waarheidstabel afgelees. Hierdie toetsvektore word in tabel 5.5 aangetoon.

Tabel 5.5: Moontlike toetsvektore vir foute op nodes H tot J van generator.

Toets	Kondisie	Moontlike toetsvektore
H/0	H=1	t1, t2, t5, t6, t8, t11, t12, t15
H/1	H=0	t0, t3, t4, t7, t9, t10, t13, t14
I/0	I=1	t1, t3, t4, t6, t8, t10, t13, t15
I/1	I=0	t0, t2, t5, t7, t9, t11, t12, t14
J/0	J=1	t2, t3, t4, t5, t8, t9, t14, t15
J/1	J=0	t0, t1, t6, t7, t10, t11, t12, t13

iii) Bepaling van minimum toetsstel

Vanuit die 16 moontlike toetsvektore moet 'n minimum toetsstel bepaal word. 'n Metode van inspeksie van tabelle 5.2, 5.4 en 5.5 is gebruik vir die bepaling van die minimum toetsstel. Eerstens word alle essensiële-toetse in die minimum toetsstel ingesluit (indien sulke toetse bestaan).

T3 kan as die eerste toetsvektor in die minimum toetsstel geneem word. Alle foute wat deur t3 getoets word kan van die foutlys verwyder word. Addisionele toetsvektore word gekies totdat alle foute deur ten minste een toetsvektor getoets word. T3, t7, t10 en t12 vorm 'n minimum toetsstel. Die minimum toetsstel (horisontale-as) word saam met die moontlike foute (vertikale-as) in 'n foutmatriks in tabel 5.6 aangetoon.

Indien die foutlys nagegaan word sal gesien word dat ononderskeidbare foute voorkom (f1 en f8, f2 en f7, ens.). Addisionele toetse moet bygevoeg word om onderskeid te tref tussen sulke foute. 'n Addisionele toets wat bygevoeg kan word is t15 waarna alle ononderskeidbare foute verwyder is.

iv) Foutdiagnosering

Vanaf die foutmatriks word foutdiagnosering gedoen deur tegnieke soos bespreek in afdeling 2.8. In plaas daarvan om 'n vloeidiagram te teken word die resultate direk in tabelvorm aangeteken, wat dan net

Tabel 5.6: Foutmatriks van generator

	Minimum toetsstel				Addisionele toetse
	t3	t7	t10	t12	t15
A/0 (f1)	X	X			X
A/1 (f2)			X	X	
B/0 (f3)	X	X	X		X
B/1 (f4)				X	
C/0 (f5)		X		X	X
C/1 (f6)	X		X		
D/0 (f7)			X	X	X
D/1 (f8)	X	X			
E/0 (f9)			X		
E/1 (f10)	X	X		X	X
F/0 (f11)	X			X	
F/1 (f12)		X	X		X
G/0 (f13)	X		X	X	
G/1 (f14)		X			X
H/0 (f15)				X	X
H/1 (f16)	X	X	X		
I/0 (f17)	X		X		X
I/1 (f18)		X		X	
J/0 (f19)	X				X
J/1 (f20)		X	X	X	

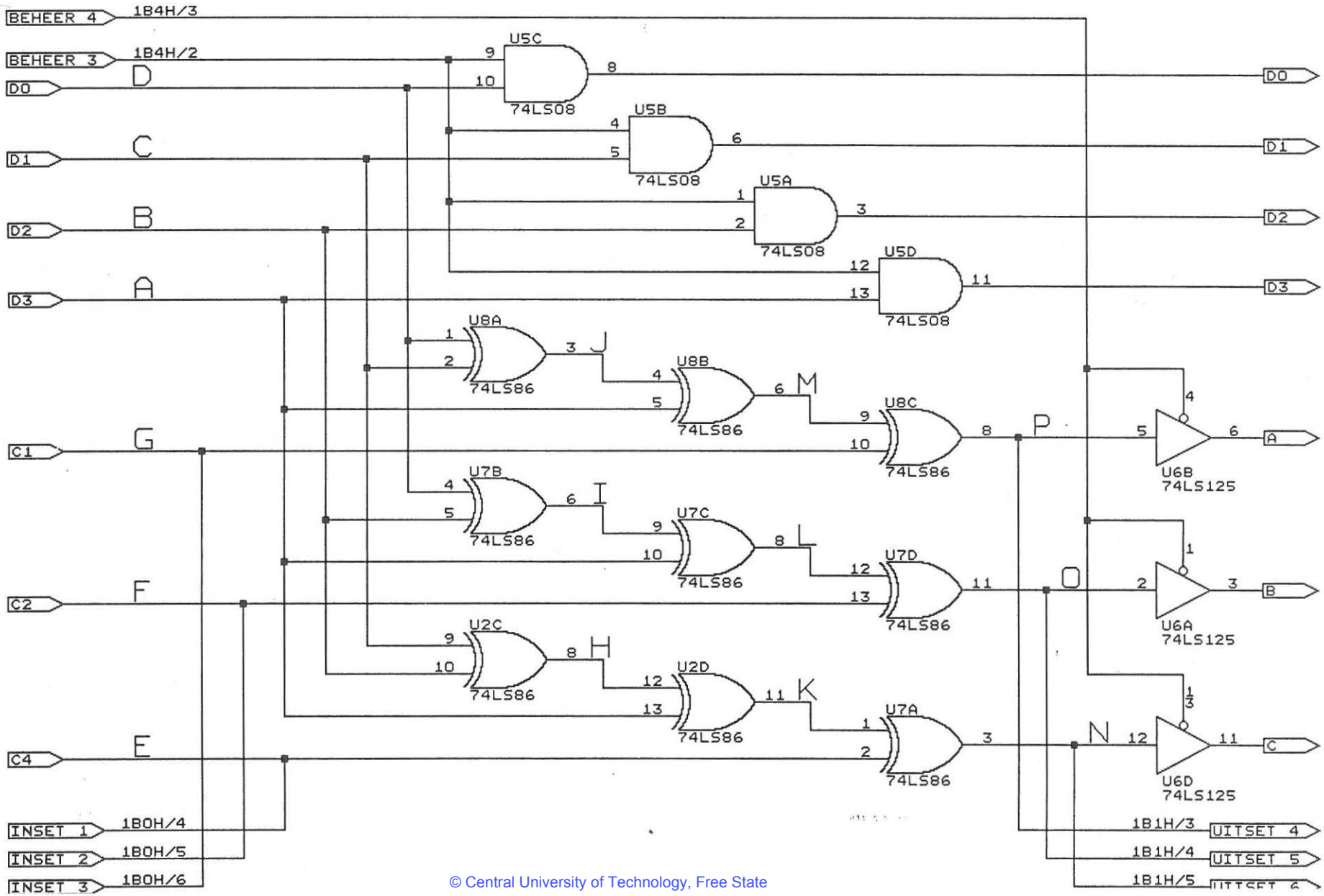
so saam met beheerpoorte, inset- en uitsetpoorte, ens. in 'n lêer ingevoer kan word. Tabel 5.7 toon die resultaat.

Tabel 5.7: [illegible]ing vanaf foutmatriks

Toetsnommer	Vektor	Resultaat	Slaag	Faal
1t3	3	6	1t7	2t7
1t7	7	0	1t10	3t10
1t10	10	2	1t12	4t12
1t12	12	1	1t15	5t15
1t15	15	7	f0	
2t7	7	0	2t10	6t10
2t10	10	2	2t12	7t12
2t12	12	1	f19	f11
3t10	10	2	3t12	8t12
3t12	12	1	f14	9t15
4t12	12	1	f9	10t15
5t15	15	7	f4	f15
6t10	10	2	6t12	11t15
6t12	12	1	6t15	f10
6t15	15	7	f8	f1
7t12	12	1	7t15	f13
7t15	15	7	f6	f17
8t12	12	1	f12	f20
9t15	15	7	f18	f5
10t15	15	7	f2	f7
11t15	15	7	f16	f3

5.2.2.2 Foutwoordbepaler

Die foutwoordbepaler word in figuur 5.3 aangetoon. Net soos by die generator word addisionele hekke bygevoeg om die kring te isoleer. Insette 1 tot 3 word aan bisse 4 tot 6 van poort 1B0H (432) verbind



terwyl uitsette 4 tot 6 aan bisse 3 tot 5 van poort 1B1H (433) verbind word. Beheerleidings 3 en 4 word onderskeidelik aan bisse 2 en 3 van poort 1B4H (436) verbind. Die beheerwoord verander met die toetsing van die foutwoordbepaler na 0BH (11). Die maskeergreep van "I_PRT" verander nou na 38H (56) om voorsiening te maak vir die nuwe uitsette wat gelees moet word.

i) Bepaling van die foutlys

Om die nodes maklik uit te ken word die nodes genommer vanaf A tot P. Die foutlys is elke node vasgeklem-by-0 en vasgeklem-by-1.

ii) Bepaling van die toetsvektore

Met die vorige kring het die waarheidstabel slegs 2^4=16 moontlike kombinasies gehad. Die foutwoord-bepaler sal egter 2^7=128 moontlike kombinasies hê. 'n Waarheidstabel is dus nie prakties in dié geval nie. Om 'n minimum toetsstel vanuit die staanspoor te bepaal, word elke inset om die beurt gelyk aan 1 gestel met al die ander insette gelyk aan 0. Die toetsstel wat so bepaal word, word saam met die foute in 'n foutmatriks in tabel 5.8 getoon.

Indien tabel 5.8 nagegaan word, word gevind dat geen ononderskeidbarefoute voorkom nie en dus kan die toetse net so as die minimum toetsstel aanvaar word.

Tabel 5.8: Foutmatriks van foutwoordbepaler

	t1	t2	t4	t8	t16	t32	t64
A/0 (f1)	X						
A/1 (f2)		X	X	X	X	X	X
B/0 (f3)		X					
B/1 (f4)	X		X	X	X	X	X
C/0 (f5)			X				
C/1 (f6)	X	X		X	X	X	X
D/0 (f7)				X			
D/1 (f8)	X	X	X		X	X	X
E/0 (f9)					X		
E/1 (f10)	X	X	X	X		X	X
F/0 (f11)						X	
F/1 (f12)	X	X	X	X	X		X
G/0 (f13)							X
G/1 (f14)	X	X	X	X	X	X	
H/0 (f15)		X	X				
H/1 (f16)	X			X	X	X	X
I/0 (f17)	X		X				
I/1 (f18)		X		X	X	X	X
J/0 (f19)	X	X					
J/1 (f20)			X	X	X	X	X
K/0 (f21)		X	X	X			
K/1 (f22)	X				X	X	X
L/0 (f23)	X		X	X			
L/1 (f24)		X			X	X	X
M/0 (f25)	X	X		X			
M/1 (f26)			X		X	X	X
N/0 (f27)		X	X	X			X
N/1 (f28)	X				X	X	

Tabel 5.8 ks van foutwoordbepaler (vervolg)

	t1	t2	t4	t8	t16	t32	t64
O/0 (f29)	X		X	X		X	
O/1 (f30)		X			X		X
P/0 (f31)	X	X		X	X		
P/1 (f32)			X			X	X

iii) Foutdiagnosering

Foutdiagnosering word weereens vanaf die foutmatriks gedoen en die resultate word in tabel 5.9 aangeteken.

5.2.2.3 Korrigeerder

Die korrigeerder word in figuur 5.4 aangetoon. Om die werking van 'n toetsstel beter te illustreer is die kring nie optimaal vereenvoudig nie maar soos in figuur 5.4 gebou. Die nodes in die kring word genommer vanaf A tot X, D0 tot D3 en Y3, Y5, Y6 en Y7.

'n Beheerwoord van 0FH (15) word na poort 1B4H (436) gestuur. Dit sal die EN-hekke (op die dataleidings) aanskakel en die drietoestandbuffers afskakel in beide die generator en die foutwoordbepaler (sien

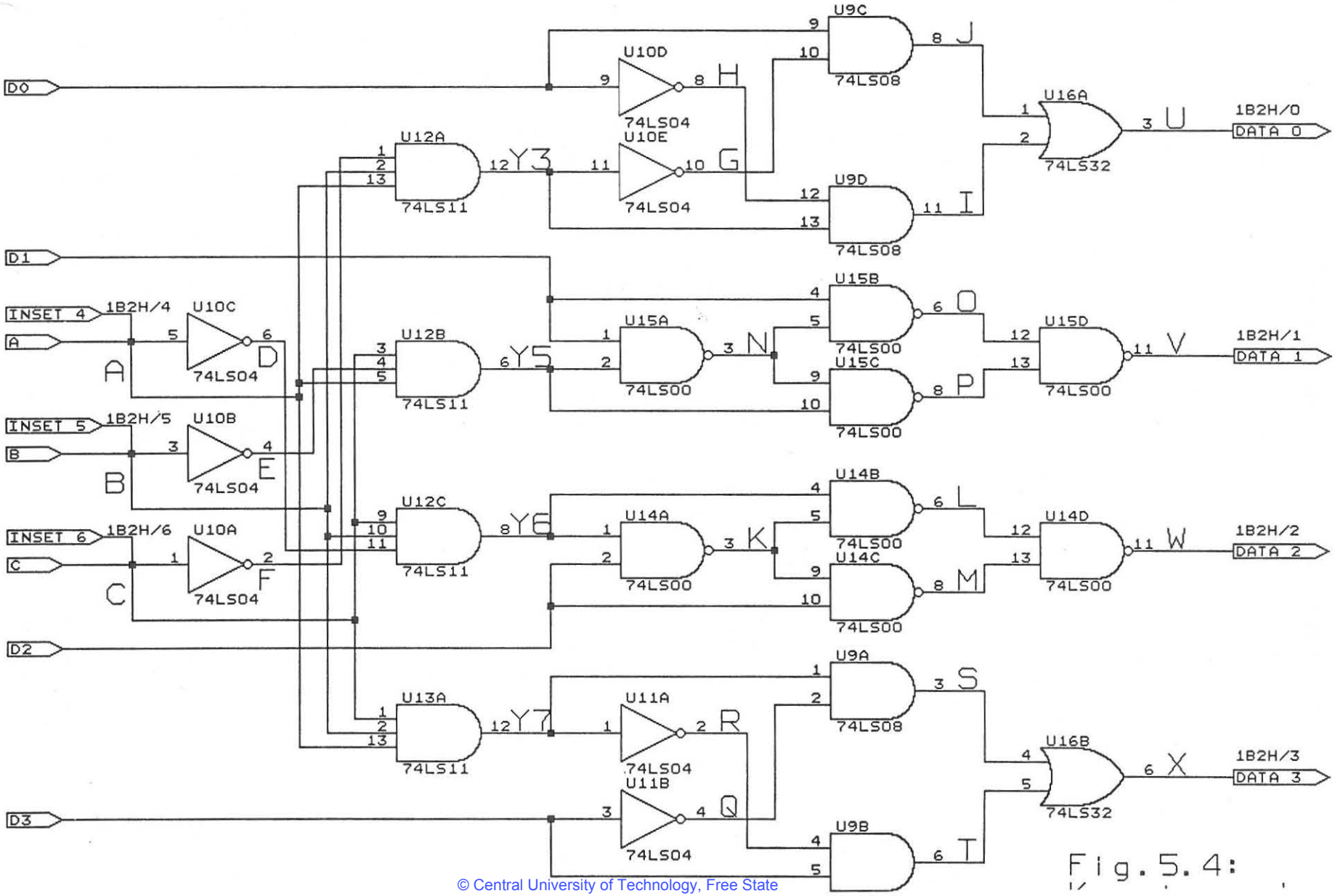


Fig. 5.4:

Tabel 5.9: Foutdiagnosering van foutwoordbepaler

Toetsnommer	Vektor	Resultaat	Slaag	Faal
1t1	1	24	1t2	2t2
1t2	2	40	1t4	3t4
1t4	4	48	1t8	4t8
1t8	8	56	1t16	f1
1t16	16	8	1t32	f13
1t32	32	16	1t64	f11
1t64	64	32	f0	f9
2t2	2	40	2t4	5t4
2t4	4	48	2t8	6t8
2t8	8	56	2t16	f16
2t16	16	8	f7	7t64
3t4	4	48	3t8	8t8
3t8	8	56	3t16	f18
3t16	16	8	f5	9t32
4t8	8	56	4t16	f20
4t16	16	8	4t32	f26
4t32	32	16	f3	f32
5t4	4	48	5t8	10t8
5t8	8	56	f19	11t16
6t8	8	56	f17	12t16
7t64	64	32	f28	f22
8t8	8	56	f15	13t16
9t32	32	16	f30	f24
10t8	8	56	f2	14t16
11t16	16	8	f25	15t32
12t16	16	8	12t32	f6
12t32	32	16	f23	f29
13t16	16	8	13t64	f8

Tabel 5.9: ring van foutwoordbepaler (vervolg)

Toetsnommer	Vektor	Resultaat	Slaag	Faal
13t64	64	32	f21	f27
14t16	16	8	f14	16t32
15t32	32	16	f31	f4
16t32	32	16	f12	f10

figure 5.2 en 5.3). Vir die toetsing van die korrigeerder word twee rekords benodig vir uitset poorte ("O_PRT). D0 tot D3 (figuur 5.4) is aan bisse 0 tot 3 van poort 1B0H (432) verbind terwyl insette 4 tot 6 (figuur 5.4) aan bisse 4 tot 6 van poort 1B2H (434) verbind is. Die uitset van die kring (DATA 0 tot DATA 3 in figuur 5.4) word aan inset poort ("I_PRT) 1B2H (434) verbind.

i) Foutineenstorting

Die aantal moontlike foute kan verminder word deur gebruik te maak van foutineenstorting. Die ononderskeidbare- en dominantefoute in die kring word vervolgens gelys:

Ononderskeidbarefoute	Dominantefoute
D/0 ; Y6/0	**D/1 -> Y6/1**
E/0 ; Y5/0	**E/1 -> Y5/1**
F/0 ; Y3/0	**F/1 -> Y3/1**
G/0 ; J/0	**G/1** -> J/1
H/0 ; I/0	**H/1 -> I/1**
J/1 ; **I/1** ; U/1	J/0, I/0 -> **U/0**
L/0 ; M/0 ; W/1	**L/1, M/1 -> W/0**

Ononderskeidbare foute	Dominantefoute (vervolg)
O/0 ; P/0 ; V/1	O/1, P/1 -> V/0
Q/0 ; S/0	Q/1 -> S/1
R/0 ; T/0	R/1 -> T/1
S/1 ; T/1 ; X/1	S/0, T/0 -> X/0

ii) Bepaling van foutlys

Uit die lys van ononderskeidbare- en dominantefoute word die foute in vetdruk behou. Die res van die foute word deur ander toetsvektore geïdentifiseer en kan dus uit die foutlys weggelaat word. Alhoewel sommige van die dominantefoute ook weggelaat kan word, word dié foute soos aangetoon wel in die foutlys ingesluit aangesien beter foutdiagnosering so verkry word. Met foutineenstorting in ag geneem kan die foutlys as volg saamgestel word:

A/0	(f1)	G/0, J/0	(f26)
A/1	(f2)	G/1	(f27)
B/0	(f3)	H/0, I/0	(f28)
B/1	(f4)	H/1	(f29)
C/0	(f5)	I/1, J/1, U/1	(f30)
C/1	(f6)	K/0	(f31)
D/0, Y6/0	(f7)	K/1	(f32)
D/1	(f8)	L/0, M/0, W/1	f33)
E/0, Y5/0	(f9)	L/1	(f34)
E/1	(f10)	M/1	(f35)
F/0, Y3/0	(f11)	N/0	(f36)
F/1	(f12)	N/1	(f37)
Y3/1	(f13)	O/0, P/0, V/1	(f38)
Y5/1	(f14)	O/1	(f39)

Y6/1	(f15)	P/1	(f40)
Y7/0	(f16)	Q/0, S/0	(f41)
Y7/1	(f17)	Q/1	(f42)
D0/0	(f18)	R/0, T/0	(f43)
D0/1	(f19)	R/1	(f44)
D1/0	(f20)	S/1, T/1, X/1	(f45)
D1/1	(f21)	U/0	(f46)
D2/0	(f22)	V/0	(f47)
D2/1	(f23)	W/0	(f48)
D3/0	(f24)	X/0	(f49)
D3/1	(f25)		

iii) Bepaling van toetsvektore

Alle moontlike toetsvektore word nou vir elke fout in die foutlys bepaal. Uit hierdie toetsvektore word dan 'n minimum toetsstel saamgestel. Aangesien baie toetsvektore bestaan vir elke moontlike fout is besluit om insette D0 tot D3, indien nie van toepassing op 'n spesifieke toets nie, gelyk aan 0 te maak. So byvoorbeeld sal D1 tot D3 gelyk aan 0 gestel word indien die kringgedeelte met Y3 as inset en U as uitset getoets word. Die toetsvektore vir die foute in die vorige foutlys word in tabel 5.10 gegee.

iv) Bepaling van minimum toetsstel

Vanuit al hierdie toetsvektore kan 'n minimum toetsstel saamgestel word. Dit word gedoen deur ten eerste die essensiële-toetse in die toetsstel in te

Tabel 5.10: vir foute van korrigeerder

Fout	Moontlike toetsvektore
f1	t48, t49, t80, t82, t112, t116, t120
f2	t32, t33, t64, t66, t96, t100, t104
f3	t48, t96, t100, t112, t114, t120
f4	t16, t17, t64, t68, t80, t82, t88
f5	t80, t82, t96, t100, t112, t113, t120
f6	t16, t18, t36, t48, t49, t56
f7	t96, t100
f8	t112, t116
f9	t80, t81
f10	t112, t114
f11	t48, t49
f12	t112, t113
f13	t0, t1, t16, t17, t32, t33, t64, t65 t80, t81, t96, t97, t112, t113
f14	t0, t2, t16, t18, t32, t34, t48, t50 t64, t66, t96, t98, t112, t114
f15	t0, t4, t16, t20, t32, t36, t48, t52, t64, t68, t80, t84, t116, t120
f16	t112, t120
f17	t0, t8, t16, t24, t32, t40, t48, t56, t64, t72, t80, t88, t96, t104
f18	t1, t17, t33, t49, t65, t81, t97, t113
f19	t0, t16, t32, t48, t64, t80, t96, t112
f20	t2, t18, t34, t50, t66, t82, t98, t114
f21	t0, t16, t32, t48, t64, t80, t96, t112
f22	t4, t20, t36, t52, t68, t84, t100, t116
f23	t0, t16, t32, t48, t64, t80, t96, t112
f24	t8, t24, t40, t56, t72, t88, t104, t120

Tabel 5.10: vir foute van korrigeerder (vervolg)

Fout	Moontlike toetsvektore
f25	t0, t16, t32, t48, t64, t80, t96, t112
f26	t1, t17, t33, t65, t81, t97, t113
f27	t49
f28	t48
f29	t49
f30	t0, t16, t32, t49, t64, t80, t96, t112
f31	t4, t20, t36, t52, t68, t84, t96, t116
f32	t100
f33	t0, t16, t32, t48, t64, t80, t100, t112
f34	t4, t20, t36, t52, t68, t84, t116
f35	t4, t20, t36, t52, t68, t84, t116
f36	t2, t18, t34, t50, t66, t80, t98, t114
f37	t82
f38	t0, t16, t32, t48, t64, t82, t96, t112
f39	t2, t18, t34, t50, t66, t98, t114
f40	t80
f41	t8, t24, t40, t56, t72, t88, t104
f42	t120
f43	t112
f44	t120
f45	t0, t16, t32, t48, t64, t80, t96, t120
f46	t1, t17, t33, t48, t65, t81, t97, t113
f47	t2, t18, t34, t50, t66, t80, t98, t114
f48	t4, t20, t36, t52, t68, t84, t96, t116
f49	t8, t24, t40, t56, t72, t88, t104, t112

sluit. Die volgende is essensiële-toetse (met aanduiding van die betrokke fout):

f28	t48
f27, f29	t49
f40	t80
f37	t82
f32	t100
f43	t112
f42, f44	t120

Foute wat nie gedek word nie en waarvoor nog addisionele toetsvektore gekies moet word is die volgende:

f26, f31, f34, f35, f39, f41, en f48

Die toetsvektore wat gekies is vir hierdie foute is:

t2, t17, t68 en t104

Die foutdekking van elke toets in die minimum toetsstel word vervolgens bepaal aangesien uitwaaiering veroorsaak dat waardes by die inset van die kring deur die hele kring versprei. Bykans enige uitset kan dus verander indien 'n fout in die kring voorkom. As gevolg van die geweldige aantal moontlike toetsvektore is dit beter om foutdekking van die minimum toetsstel te bepaal. Die foutdekking word in tabel 5.11 aangetoon. Let op die addisionele foute wat deur sommige toetsvektore geïdentifiseer word met inagneming van foutdekking.

Tabel 5.11: Roetmatriks van korrigeerder

	t2	t17	t48	t49	t68	t80	t82	t100	t104	t112	t120
f1			X	X		X	X			X	X
f2					X			X	X		
f3			X	X				X	X	X	X
f4		X			X	X	X				
f5						X	X	X	X	X	X
f6		X	X	X							
f7								X	X		
f8										X	X
f9						X	X				
f10										X	X
f11			X	X							
f12										X	X
f13	X	X			X	X	X	X	X	X	X
f14	X	X	X	X	X			X	X	X	X
f15	X	X	X	X	X	X	X			X	X
f16										X	X
f17	X	X	X	X	X	X	X	X	X		
f18		X		X							
f19	X		X		X	X	X	X	X	X	X
f20	X						X				
f21		X	X	X	X	X		X	X	X	X
f22					X			X			
f23	X	X	X	X		X	X		X	X	X
f24									X		X
f25	X	X	X	X	X	X	X	X		X	
f26		X									
f27				X							
f28			X								

Tabel 5.11: Foutmatriks van korrigeerder (vervolg)

	t2	t17	t48	t49	t68	t80	t82	t100	t104	t112	t120
f29				X							
f30	X			X	X	X	X	X	X	X	X
f31					X				X		
f32								X			
f33	X	X	X	X		X	X	X		X	X
f34									X		
f35					X						
f36	X					X					
f37							X				
f38		X	X	X	X		X	X	X	X	X
f39	X										
f40						X					
f41										X	
f42											X
f43									X		
f44											X
f45	X	X	X	X	X	X	X	X			X
f46		X	X								
f47	X					X					
f48					X				X		
f49									X	X	

v) Foutdiagnosering

Foutdiagnosering word weereens vanaf die foutmatriks gedoen en in lêervorm geskryf (sien BYLAAG E).

5.2.2.4 Addisionele toetse

Toetsvektore is bepaal vir al die foute soos in die foutlys uiteengesit. Die enigste oorblywende foute is foute wat op die beheerleidings mag voorkom. Tydens die uitvoering van die toetsvektore - soos uiteengesit in afdelings 5.2.2.1 tot 5.2.2.3 - is sommige foute op die beheerleidings geïdentifiseer. Om die oorblywende foute te kan geïdentifiseer is hierdie foute een na die ander in die kring veroorsaak waarna die toetsstel uitgevoer is. Foutinligting wat so verkry is, is by die toetsstel bygewerk. Addisionele toetsvektore is bereken vir die foute wat nie deur middel van hierdie tegniek geïdentifiseer is nie (kring toets foutvry indien die bestaande toetsstel op die kring uitgevoer word).

Dieselfde tegniek is ook gebruik vir die bepaling van foutinligting vir bruggingsfoute. Deur 'n bruggingsfout in die kring te veroorsaak en die toetsstel toe te pas op die kring is die nodige foutinligting verkry wat dan in die toetsstel ingesluit word.

5.2.3 Resultate

Die toetsvektore - soos bepaal vir die generator, foutwoordbepaler en korrigeerder - is gekombineer as 'n toetsstel en word aangetoon in BYLAAG E. Addisionele

toetse soos bep[illegible].4 word ook in die toetsstel aangetoon. Om die effektiwiteit van die toetsstel te bepaal is elke fout wat in die foutlys voorkom op die bord gesimuleer waarna die toetsstel op die bord uitgevoer is. Alle foute wat gesimuleer is, is korrek deur die OTS geïdentifiseer tydens die uitvoering van die toetsstel daarop.

5.3 MSI logika

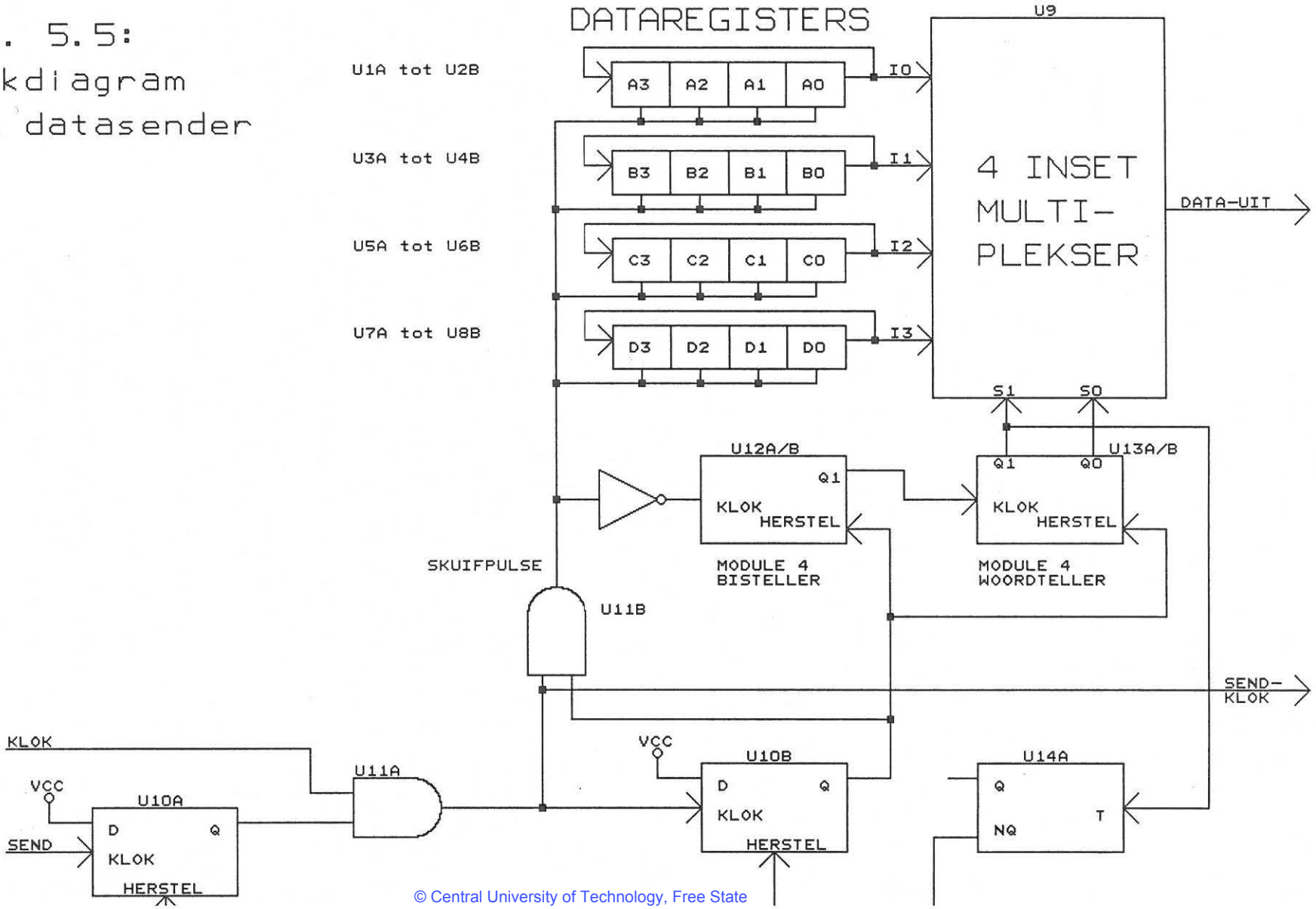
Die tweede en finale kring is ontwerp rondom die gebruik van multipleksers, kombinasielogika en 'n verskeidenheid wip-kringe.

5.3.1 Ontwerp van die kring

'n Sinkrone-datatransmissiestelsel, wat gebruik word om vier 4-bis datawoorde vanaf 'n sender na 'n ontvanger te stuur, is ontwerp en gebou. Vir die evaluering van die toetsstelsel is slegs die sender - soos blokdiagram-maties in figuur 5.5 aangetoon - gebruik.

Die vier datawoorde word in skuifregisters A tot D (U1 tot U8) gestoor. Die uitset van elke registerbank word terugverbind na die inset van dieselfde registerbank. Die mins-beduidende bis van elke register vorm die vier insette na die multiplekser (U9). Elke register sal regs skuif met 'n leirand wat by die uitset van EN-hek U11B verkry word. Data word in serie versend vanaf uitset-1Y (DATA-UIT) van die multiplekser.

Fig. 5.5:
Blokdiagram
van datasender



Die twee module-4 tellers (U12A en B en U13A en B) beheer die versending van die data vanaf die dataregisters na die uitset van die multiplekser. Die woordteller (U13A en B) kies die verlangde dataregister terwyl die bisteller (U12A en B) verseker dat die vier bisse van elke register versend word. Die bisteller skuif een telling aan met elke skuifpuls en tel dus vanaf 00 tot 11 en terug na 00 in vier skuifpulse. Die volgrand (elke vierde puls) by die uitset van die bisteller laat die woordteller een telling aanstap, wat tot gevolg het dat die volgende dataregister na die uitset van die multiplekser deurverbind word. Volgens hierdie metode word vier bisse van elke register - beginnend by register A - bis vir bis in serie na die uitset van die multiplekser gestuur.

U10A en U10B word gebruik vir die beheer van oorsending van die data. Normaalweg sal die uitsette van genoemde wipkringe LAAG wees. Die LAAG by die uitset van U10A verhoed dat klokpulse deur EN-hek U11A beweeg terwyl die LAAG by die uitset van U10B verseker dat beide die tellers by 00 gehou word.

Met die uitset van die woordteller gelyk aan 00 word die bis wat in A0 gestoor word na die uitset van U9 gestuur. Indien 'n sendpuls nou toegepas word op U10A sal dit Q stel wat tot gevolg het dat die klokpuls deur U11A gestuur word. Die klokpulse by die uitset van U11A word ook, tesame met die data, as 'n sendklok na die ontvanger gestuur. Die leirand van die eerste klokpuls

sal veroorsaak dat U10B stel. Die HOOG by die uitset van U10B verwyder die LAAG vanaf die herstelleiding van die tellers en veroorsaak terselfdertyd dat klokpulse deur U11B gestuur kan word. Die pulse by die uitset van U11B word as skuifpulse op die registers gebruik asook as klokpulse vir die bisteller.

Wanneer die volgrand op Q1 (woordteller) voorkom aan die einde van die siklus genereer monostabiele multivibrator (U14A) 'n puls wat U10A en B herstel. Dit het tot gevolg dat klokpulse geblokkeer word deur U11A en dat beide die tellers tot 00 sal terugkeer. Die volledige kringdiagram word in figure 5.6(a) en (b) aangetoon.

5.3.2 Bepaling van die foutlys

Die vasgeklem-by-waarde model is gebruik vir die bepaling van die foutlys. Elke node in die kring kan dus vasgeklem wees by 0 of 1. Slegs nodes wat nodig is vir die normale werking van die kring is ingesluit in die foutlys. So byvoorbeeld is die Q-uitset van U14A in figuur 5.6(b) nie in die foutlys ingewerk nie. Nodes wat egter vir toetsing gebruik is - soos byvoorbeeld "CLR" van U14A - is bygewerk.

5.3.3 Bepaling van 'n toetsstel

As gevolg van die besondere ontwerp van die kring kan die prosedure wat by die eerste kring gevolg is vir die bepaling van 'n toetsstel nie gebruik word by hierdie

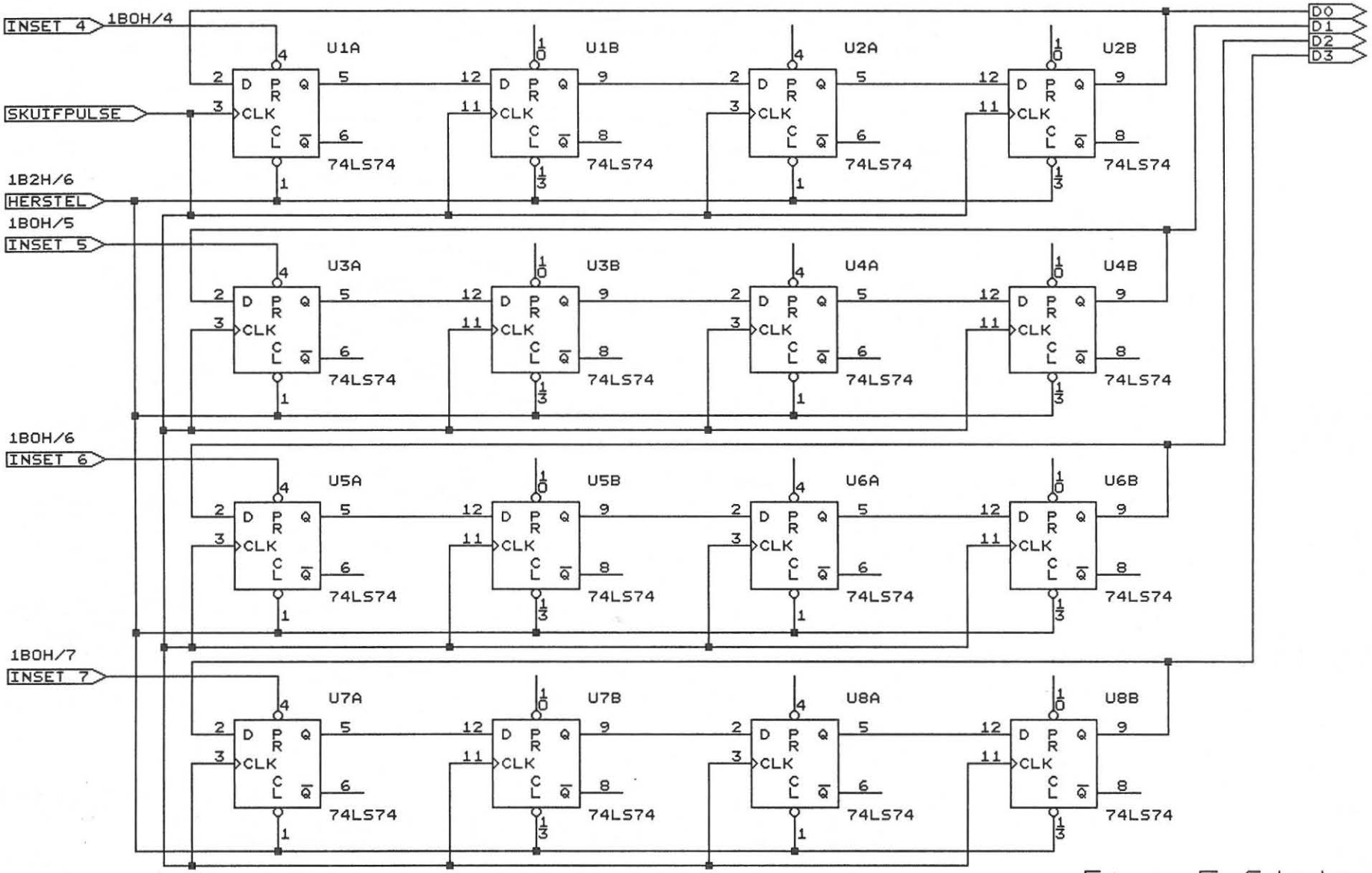


Fig. 5.6(a):

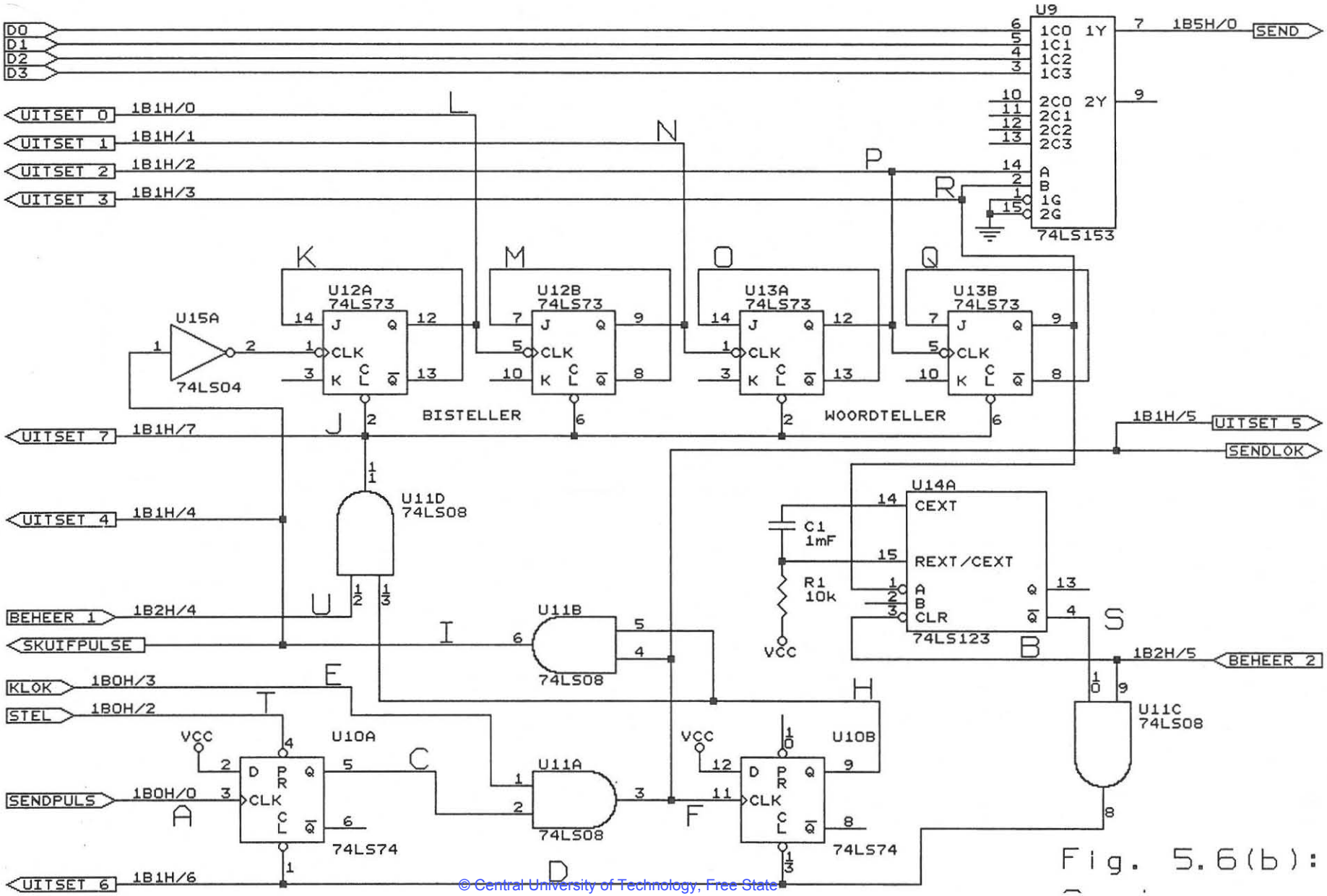


Fig. 5.6(b):

kring nie. Dit is na aanleiding van die feit dat die kring op enige stadium in 'n sekere staat verkeer as gevolg van die klokpulse wat reeds daarop uitgevoer is. Funksies soos die herstel van die insetkring en die bis- en woordtelling moet dus 'n aantal kere uitgevoer word gedurende die toetsing van die kring.

Ontwerp-om-te-kan-toets tegnieke is weereens gebruik om die bepaling van die toetsvektore te vereenvoudig. Addisionele beheer- en waarnemingspunte is gebruik (soos aangedui in figuur 5.6(b)) terwyl twee EN-hekke (U11C en U11D) ook in die kring bygevoeg is. U11C is bygevoeg aangesien die monostabiele multivibrator (U14A) gebruik word om U10A en U10B te herstel en dit wenslik is om die herstelfunksie te verifieer, voordat hierdie funksie gebruik word om die kring in 'n bekende staat te plaas. U11D is bygevoeg om die herstelleiding van wipkringe U12A tot U13B te verifieer asook om die kring op te deel in toetsbare eenhede.

Vir die toetsing van hierdie kring is die twee 8255 PRKs soos volg opgestel (slegs poorte wat gebruik is word aangetoon):

1B0H	Uitset
1B1H	Inset
1B2H	Uitset (Hoë hap)
1B5H	Inset

Die toetsing van die sender in figuur 5.6 (a) en (b) word deur middel van 'n vloeidiagram in figuur 5.7 uiteengesit. Elk van die toetse - soos in die vloeidiagram aangedui - word meer breedvoerig bespreek.

5.3.3.1 Toetsing van die monostabiele multivibrator herstelfunksie.

Alhoewel herstel (CLR) van U14A nie in die normale werking van die kring gebruik word nie, is hierdie leiding soos aangetoon in figuur 5.6(b) bedraad vir toetsdoeleindes. Vir die toetsing van die herstel-funksie word BEHEER 2 en UITSET 6 onderskeidelik as inset en uitset gebruik. Die stappe benodig vir die toetsing van die herstelfunksie van U14A word deur die vloeidiagram in figuur 5.8 aangetoon. Gedurende die uitvoering van hierdie toetsvektore word STEL (1B0H/2) by sy onaktiewe staat (HOOG) gehou.

Genoemde toetsvektore word in BYLAAG F as rekordnommers 1 tot 15 aangetoon. Foutnommers word in figuur 5.8 in hakkies na die foutinligting gegee vir kruisverwysing na verdere inligting in die toetsstel in BYLAAG F. Die herstelfunksie van U14A kan nou, aangesien dit foutvry bewys is, gebruik word gedurende die uitvoering van die res van die toetse.

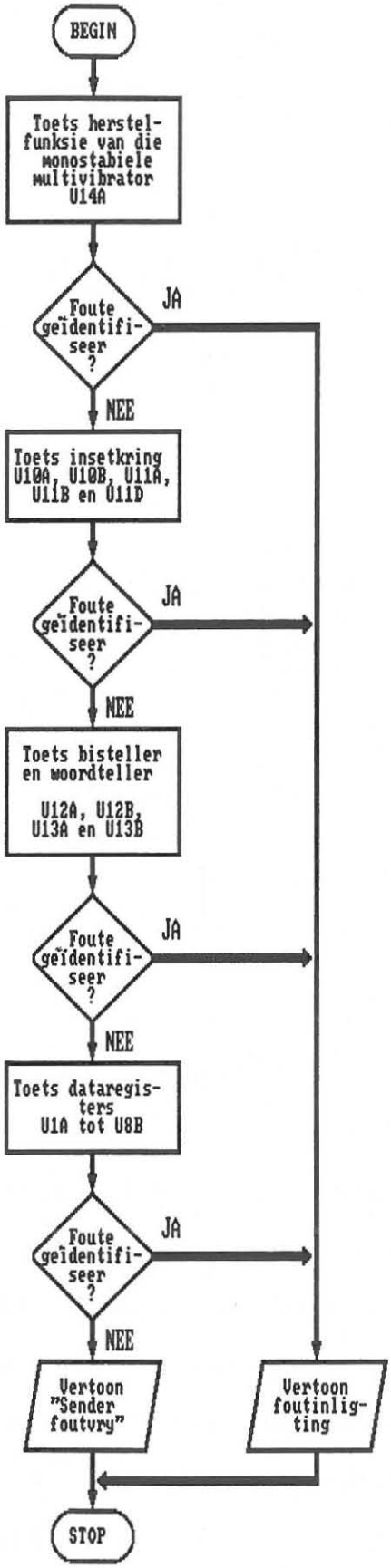


Fig. 5.7: Vloeidiagram vir die toetsing van die sender

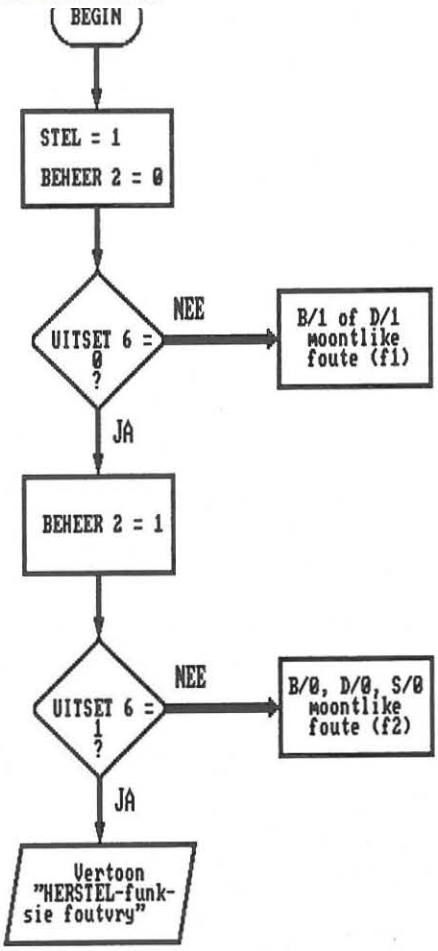


Fig. 5.8: Vloeidiagram vir die toetsing van die herstelfunksie.

5.3.3.2 Toetsing van insetkring

Die toetsing van die insetkring behels die toetsing van U10A, U10B, U11A, U11B en U11D. Hierdie toetse word in BYLAAG F as rekordnommers 16 tot 79 aangetoon. Bisse 4 en 5 van poort 1B1H word deur twee afsonderlike leesprosesse ingelees - eers bis 5 gevolg deur bis 4. Dit is gedoen aangesien die twee bisse afsonderlik ge-evalueer word. Indien een van hierdie uitsette onveranderd sal bly - na die toepassing van sekere toetsvektore - kan geen

foutdiagnos betrokke uitset gedoen word nie en word die uitset nie in figuur 5.9 getoon nie. Toetsvektore word op BEHEER 2 en STEL gestuur. Die stappe vir die aanvanklike toetsing van die insetkring word in figuur 5.9 aangetoon. Sendpuls (figuur 5.9) en klokpuls (figuur 5.10) inskrywings stel pulse voor wat onderskeidelik op SENDPULS en KLOK aangelê word. Aangesien elke deel individueel getoets word, word daar telkens by f1 begin vir die nommering van die foute. F0 dui telkens daarop dat die betrokke deel van die kring foutvry is.

Nadat foutdekking nagegaan is, is gevind dat sommige foute van die insetkring nie geïdentifiseer is nie (H/1, J/0, J/1, U/0 en U/1). Addisionele toetse is geskryf - rondom die gebruik van UITSET 7 (1B1H/7) - vir die identifisering van hierdie foute. Hierdie toetse word in figuur 5.10 aangetoon.

5.3.3.3 Toetsing van bis- en woordteller

Vir die toetsing van die bis- en woordteller word BEHEER 1, BEHEER 2, SENDPULS, en KLOK as insette gebruik terwyl UITSETTE 0 tot 3 as uitsette gebruik word. Die toetsing van die bis- en woordtellers word as 'n vloeidiagram in figuur 5.11 aangetoon. Weereens word uitsette wat geen diagnosering tot gevolg het nie, nie in die figuur getoon nie. Hierdie toetse kan in BYLAAG F gesien word as rekordnommers 80 tot 145.

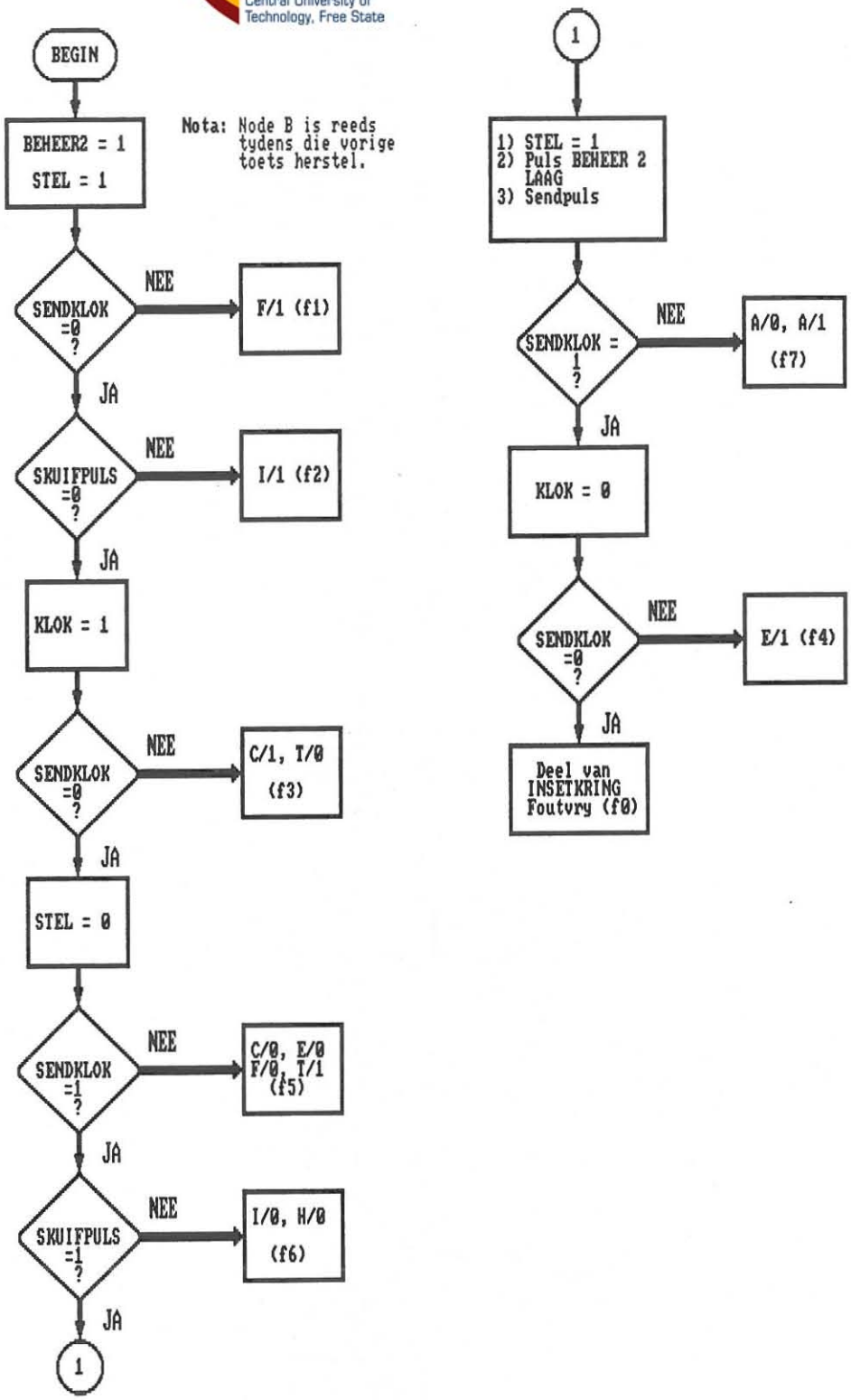


Fig. 5.9: Vloeidiagram vir die toetsing van 'n deel van die insetkring.

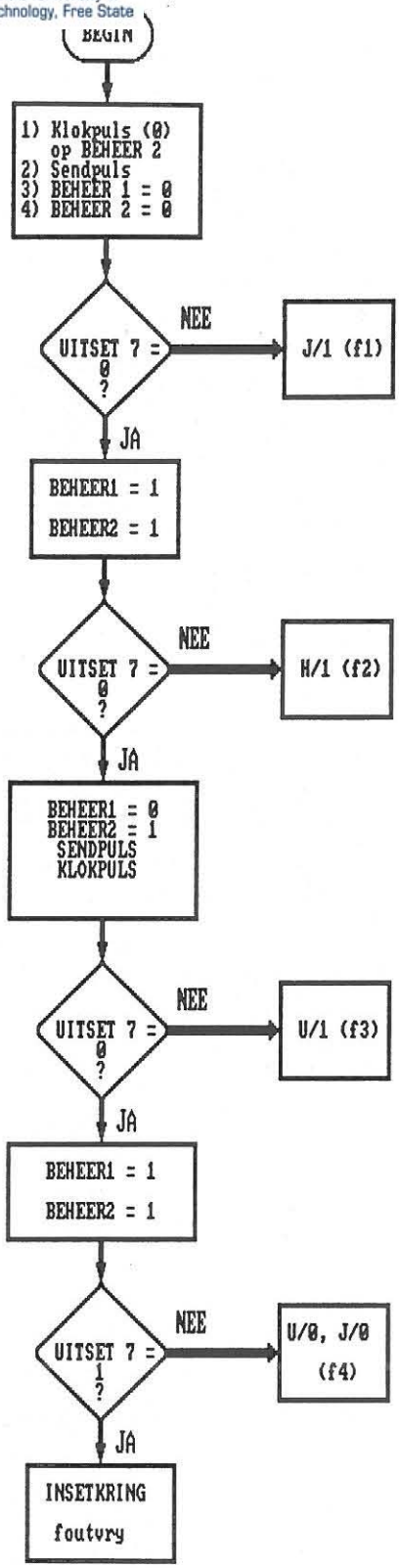


Fig. 5.10: Vloeidiagram vir die toetsing van die res van die insetkring.

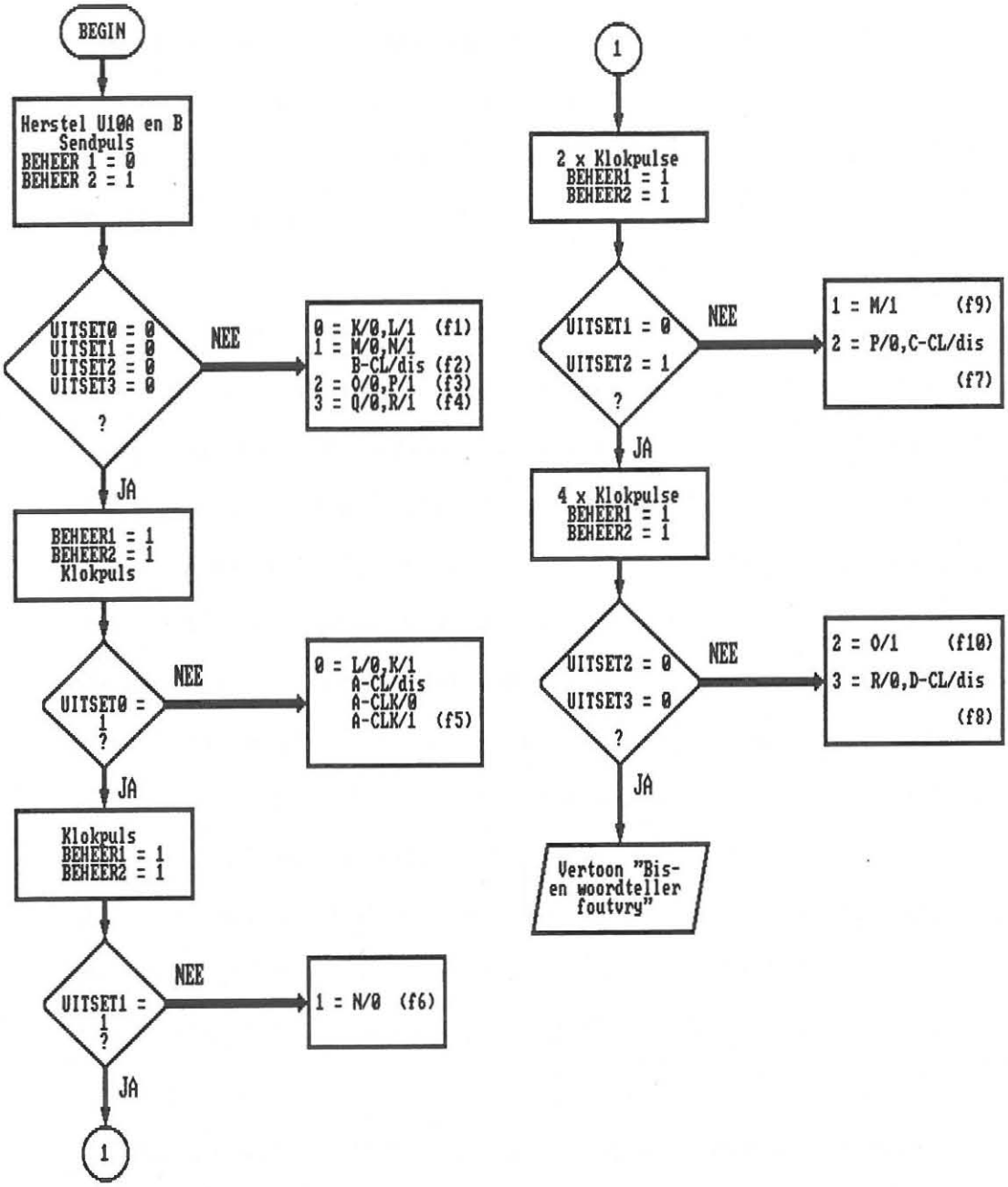


Fig. 5.11: Vloeidiagram vir die toetsing van die bis- en woordteller.

5.3.3.4 Toetsing van dataregisters

Toetsing van die dataregisters (figuur 5.6(a)) en die multiplekser (U9 in figuur 5.6(b)) word vervolgens afgehandel. Hierdie toetse word in

rekordnommers 146 tot 310 in BYLAAG F aangetoon. Toetsing van die dataregisters is gedoen deur eers vir vasgeklem-by-1 en daarna vir vasgeklem-by-0 foute te toets. Elke registerbank word in albei gevalle individueel getoets.

Toetsing van die vasgeklem-by-1 foute neem 'n aanvang deur al die registers te herstel. Deur drie klokpulse op die dataregisters toe te pas - en die uitset van die multiplekser na elke klokpuls te lees - kan vasgestel word of enige nodes in die geselekteerde dataregister by 1 vasgeklem is al dan nie. Nadat 'n registerbank getoets is word al die registers weer herstel vir die toetsing van die volgende registerbank. Die aantal klokpulse wat op die bis- en woordtellers toegepas word, moet ook in gedagte gehou word, aangesien hierdie klokpulse sal bepaal watter registerbank na die uitset van die multiplekser deurgeskakel is.

Die toetsvektore vir die toetsing van vasgeklem-by-1 foute word in tabel 5.12 aangetoon. In die tabel wys die term klok na 'n klokpuls wat op KLOK gestuur word. Die aantal klokpulse wat gestuur word, word in hakies aangedui. Sendpuls dui op 'n puls wat op SENDPULS gestuur word om U10A te stel. Die vektor wat op 1B2H gestuur word is 70H (112) aangesien BEHEER 1, BEHEER 2 en HERSTEL HOOG gehou word gedurende die uitvoering van hierdie toetse. Vir

Tabel 5.12: Toetsvektore vir die toetsing van dataregisters (vasgeklem-by-1).

Toetsnommer	1B2H	1B5H	Foutdekking
Herstel insetkring en stel daarna BEHEER 1=1 en BEHEER 2=1			
Herstel dataregisters			
Sendpuls			
t1	112	0	f1
Klok (1)			
t2	112	0	f2
Klok (1)			
t3	112	0	f3
Klok (1)			
t4	112	0	f4
Herstel insetkring			
Sendpuls			
Klok (4)			
Herstel dataregisters			
t5	112	0	f5
Klok (1)			
t6	112	0	f6
Klok (1)			
t7	112	0	f7
Klok (1)			
t8	112	0	f8
Herstel insetkring			
Sendpuls			
Klok (8)			
Herstel dataregisters			

Tabel 5.12: e vir die toetsing van dataregisters (vasgeklem-by-1) (vervolg).

Toetsnommer	1B2H	1B5H	Foutdekking
t9	112	0	f9
Klok (1)			
t10	112	0	f10
Klok (1)			
t11	112	0	f11
Klok (1)			
t12	112	0	f12
Herstel insetkring			
Sendpuls			
Klok (12)			
Herstel dataregisters			
t13	112	0	f13
Klok (1)			
t14	112	0	f14
Klok (1)			
t15	112	0	f15
Klok (1)			
t16	112	0	f16

foutinligting van die foutdekking kan rekordnommers 219 tot 254 in BYLAAG F geraadpleeg word.

Vasgeklem-by-1 foute is nou afgehandel en vasgeklem-by-0 foute word getoets. Toetsing vir vasgeklem-by-0 foute word begin deur U10A, U10B, U12A tot U13B en U1A tot U8B te herstel. 'n Klokpuls word vervolgens so saamgestel dat dit U10A sal stel asook

die stelleidings op U1A, U3A, U5A en U7A HOOG sal dryf.

Hierdie klokpulse word in rekordnommers 257 tot 264 van BYLAAG F getoon.

Toetsing vir vasgeklem-by-0 foute word gedoen deur die eerste wipkring in elke registerbank te stel, waarna klokpulse weereens toegepas word. Deur die HOOG met behulp van klokpulse deur die betrokke registerbank te stuur, word vasgestel of enige nodes by 0 vasgeklem word. Die toetsvektore word in tabel 5.13 aangetoon. Die herstel volgorde - wat uitgevoer moet word voordat 'n volgende registerbank getoets word - kan soos volg beskryf word:

Herstel insetkring

Herstel dataregisters

Sendpuls met bisse 1B0/4 tot 1B0/7 LAAG (stelfuksie vir dataregisters)

Klok (die aantal klokpulse word telkens in die tabel aangedui).

Vir foutinligting van die foute soos aangetoon word in tabel 5.13 kan rekordnommers 294 tot 308 van BYLAAG F geraadpleeg word.

Tabel 5.13: Toetsvektore vir die toetsing van dataregisters (vasgeklem-by-0).

Toetsnommer	1B2H	1B5H	Foutdekking
Herstel kring met 3 klokpulse			
t1	112	1	f1
Herstel kring met 7 klokpulse			
t2	112	1	f2
Herstel kring met 11 klokpulse			
t3	112	1	f3
Herstel kring met 15 klokpulse			
t4	112	1	f4

Alhoewel die stelfunksie van die wipkringe in figuur 5.6(a) nie gebruik word in die normale werking van die kring nie, word die vasgeklem-by-0 foute van hierdie nodes ook in die toetsstel ingesluit aangesien die genoemde defek 'n fout in die kring sal veroorsaak. Aangesien die kring met 'n lus na die inset terugverbind is - 'n HOOG word dus deur die kring gevoer - moet die hele kring herstel word voordat 'n volgende registerbank getoets word.

5.3.4 Resultate

Die toetsvektore soos saamgestel word in BYLAAG F aangetoon. Hierdie toetsvektore is die volledige toetsstel wat gebruik is vir die evaluering van die toetsstelsel. Om die effektiwiteit van die toetsstel te bepaal is elke node in die kring by logika 0 en 1 vasgeklem. Alle foute wat so gesimuleer is, is korrek

deur die OTS geïdentifiseer nadat die toetsstel daarop uitgeoefen is.

Ter illustrasie toon BYLAAG G die resultaatlêer verkry met node J vasgeklem-by-0. Uitvoering van die toetsstel het U11-11/0 en U11-12/0 as moontlike foute geïdentifiseer.

Samevatting

6.1 Belangrikheid van outomatiese toetsing

Die belangrikheid van outomatiese toetsing het duidelik na vore gekom gedurende die uitvoering van hierdie projek. Ten eerste is dit vir die vervaardiger van elektroniese kringe - net soos by enige ander produk - van uiterste belang om te verseker dat produkte aan die einde van die produksielyn korrek is. Borde wat by die gebruiker gefaal het, en wat teruggestuur word na die vervaardiger, moet getoets, foute gediagnoseer en herstel word. Hier word outomatiese toetsing benodig aangesien die vervaardiger nie die tyd om die toetse met die hand - of die opgeleide personeel om die toetse uit te voer - kan bekostig nie.

6.2 Wat behels outomatiese toetsing?

Toetsing van elektroniese kringe kan deur middel van twee afsonderlike stappe beskryf word, naamlik die identifisering van 'n fout, en die isolering van die fout. Toetsing behels die toepassing van toetsstimuli (toetsvektore) op die eenheid onder toets (EOT), die meting van die uitset van die EOT en die evaluering van die resultate wat verkry word.

'n Outomatiese-toetsstelsel bestaan uit 'n rekenaar, 'n koppelvlak tussen die rekenaar en die EOT en sagteware.

Tydens foutopsporing moet daar altyd gepoog word om die kleinste moontlike foutiewe eenheid of komponent te bepaal. Dit is gedoen deur die herhaaldelike toepassing van logikawaardes by die primêre insette en die waarneming van die resultate by die primêre uitsette van die kring. Deur die uitsette wat so verkry is te vergelyk met die verwagte foutvrye uitset van die kring - vir elke insetkondisie - is foutdiagnosering gedoen.

Elke toets - wat bestaan uit 'n insetwaarde saam met die foutvrye uitsetwaarde - staan as 'n toetspatroon of toetsvektor bekend. 'n Volledige stel toetsvektore, wat benodig word om 'n kring ten volle te kan toets, staan as 'n toetsstel bekend. Die toetsstel is so saamgestel dat indien 'n bord met 'n defek getoets word, die bord ten minste een van die toetse sal faal.

Bykans alle toetstegnieke het as uitgangspunt 'n foutmodel wat gebaseer word op tipiese foute wat in die praktyk voorkom. Die model wat gebruik is vir die doel van hierdie projek is die vasgeklem-by-waarde model. Om vir sodanige foute te toets is elke node in die kring om die beurt vanaf 'n 0 na 'n 1 en vanaf 'n 1 na 'n 0 geskakel terwyl die resultate by die uitset waargeneem is.

Die belangrikste aspek van toetspatroongenerering was om te verseker dat 'n fout wat by die inset van 'n komponent voorkom, deur die komponent verplaas word, sodat dit tydens die toetsing 'n verandering by die uitset van die komponent teweegbring. 'n Sensitiewe-pad is dus geskep vir die

verplasing van 'n fout vanaf die foutiewe node na die uitset van die kring.

Daar bestaan twee stappe indien enige node getoets word. Eerstens is die insetwaardes so gekies dat dit die betrokke node wat getoets word aktiveer. Dit staan as beheerbaarheid bekend. Tweedens moes die effek hiervan na die uitset van die EOT verplaas word. Dit staan as waarneembaarheid bekend.

Om die kringe te kan toets is daar eerstens 'n foutlys saamgestel deur gebruik te maak van die vasgeklem-by-waarde foutmodel. Vir elke fout wat in die foutlys voorkom is daar 'n toetsvektor bepaal. Vanuit al die toetsvektore wat bepaal is, is 'n minimum toetsstel saamgestel. Die minimum toetsstel is so saamgestel dat indien 'n foutiewe kring getoets word, dit ten minste een toets faal indien die toetsstel daarop uitgeoefen word. Foutdekking is bepaal vir elke toets wat in die minimum toetsstel voorkom aangesien bykans alle toetse wat geskryf is vir meer as een fout toets.

Konfliksituasies onstaan soms tydens die bepaling van toetsvektore wat tot gevolg het dat sommige foute nie getoets kan word nie. Geen konfliksituasies het egter voorgekom in die bepaling van die toetsvektore vir die twee kringe wat gebou en getoets is nie.

Die grootte van die korrigeerder se foutlys is verklein deur middel van 'n proses wat as foutineenstorting bekend staan. Foutineenstorting bestaan uit twee aksies, naamlik die identifisering van ononderskeidbare- en van dominantefoute.

Na bepaling van toetsvektore is foutdiagnosering gedoen. Dit is gedoen om die foutiewe node of komponent in die kring te identifiseer aan die hand van die resultate wat verkry is. Die foutinligting wat so verkry is - indien enige foute wel bestaan - word aan die gebruiker vertoon.

6.3 Ontwerp-om-te-kan-toets

Deur die ontwerp van 'n kring te verander kan toetsing - en dus die ontwikkeling van 'n geskikte toetsstel - vergemaklik word. Die metodes wat gebruik word om toetsbare kringe te ontwerp kan in drie kategorieë verdeel word, naamlik Ad hoc, struktureel en selftoetsing.

6.3.1 Ad hoc tegnieke

Gedurende hierdie projek is slegs van die Ad hoc tegnieke gebruik gemaak. Van die tegnieke wat gebruik is om die toetsing van die kringe te vereenvoudig is die volgende:

- Die kringe is opgedeel in kleiner eenhede wat makliker getoets kan word.
- Addisionele toetspunte is in die kringe bygevoeg.
- Lang tellers en skuifregisters is opgedeel in kleiner dele om toetsing te vergemaklik.
- Om die monostabiele multivibrator in die datasender toetsbaar te maak is die uitset as 'n toetspunt beskikbaar gestel. Beheerleidings is verskaf vir die beheer van logika wat normaalweg deur die monostabiele multivibrator aangedryf word.

6.3.2 Strukturele tegnieke

Om probleme by die opstel van 'n toetsstel vir volgordekringe te oorkom, veral by baiegrootskaalse integrasie, word van strukturele tegnieke gebruik gemaak. Van die tegnieke is die volgende:

- Vlak-sensitiewe-aftas-ontwerp (VSAO)
- Aftasbaan tegniek
- Aftas-stel logika
- Willekeurige-toegang-aftas
- Onvoltooide-aftasbaan

Alhoewel die gebruik van hierdie tegnieke in die literatuur behandel word, is dit nie in die stelsel ingesluit nie.

6.3.3 Ingeboude-selftoetsing

Ingeboude-selftoetsing is 'n tegniek wat gebruik word om 'n kring te toets sonder die gebruik van enige eksterne toetsapparaat. Toetsvektore word in die kring opgewek, toegepas en die resultate in die kring gestoor. Van die tegnieke is die volgende:

- Sinjatuuranalise
- Ingeboude-logika-blok-waarnemer

Hierdie tegnieke is nie in hierdie stelsel ingesluit nie.

6.4 Uiteensetting van die outomatiese-toetsstelsel

Na evaluering van bostaande is 'n OTS ontwikkel wat bestaan uit 'n persoonlike-rekenaar, 'n koppelvlakkaart en die hegstuk tussen die koppelvlakkaart en die EOT. Aangesien digitale borde getoets word - insette en uitsette bestaan hoofsaaklik uit 0e en 1e - word gebruik gemaak van die PC-14A koppelvlakkaart. Dit is 'n programmeerbare inset/uitset bord aanpasbaar met die IBM PC/XT/AT persoonlike-rekenaar. Die PC-14A voorsien 48 programmeerbare I/U leidings.

Die toetsstelsel maak voorsiening vir die toetsing van borde wat van 'n randkonnekteerder voorsien is. Die randkonnekteerder word gebruik vir die koppeling van die EOT en die PC-14A koppelvlakkaart.

Die sagteware van die OTS bestaan uit twee elemente, naamlik die toetsprogram self en 'n addisionele databasis vir elke bord wat deur die stelsel getoets kan word.

Aangesien 'n verskeidenheid van funksies in die sagteware van die OTS ingebou is, is die program rondom 'n menu geskryf wat die gebruik van die stelsel baie vereenvoudig. Die menu gebruik drie sub-menus wat gebruik word vir die funksies rondom die uitvoering van 'n toets, die skep van 'n nuwe lêer en die verandering van 'n bestaande lêer.

6.5 Evaluering van die OTS

Ter evaluering van die outomatiese-toetsstelsel is twee kringe gebou en 'n toetsstel vir elk bepaal. Met die eerste kring is uitsluitlik van kombinasielogika (EN-, OF-, NEN-, NOF- en eksklusiewe OF-hekke) gebruik gemaak. Met die tweede kring is gebruik gemaak van multipleksers, kombinasielogika en 'n verskeidenheid van wipkringe.

In beide kringe is alle toepaslike nodes (elke node vasgeklem-by-0 en vasgeklem-by-1) in die foutlyste ingesluit. Ter evaluering van die effektiwiteit van die toetsstelle is al die foute - soos in die betrokke foutlyste uiteengesit - op die kringe gesimuleer. Alle foute wat gesimuleer is, is wel tydens toetsing deur die OTS geïdentifiseer. Aangesien alle foute geïdentifiseer is kan gesê word dat 100% foutdekking by albei kringe behaal is.

6.6 Afsluiting

Met die toenemende verhoging in die kompleksiteit van elektroniese kringe is dit duidelik dat dit nie net die ontwerp van kringe is wat belangrik is nie maar ook die toetsing van die kringe. Aangesien tegnieke beskikbaar is om die ontwerp van 'n kring te verander - wat beter en eenvoudiger toetsing tot gevolg sal hê - is dit duidelik dat toetsing vanaf die beginstadium 'n integrale deel van die ontwerpproses moet uitmaak. Bestaande kringe wat in groot hoeveelhede gebruik word kan herontwerp word om outomatiese toetsing moontlik te maak. Deur sulke veranderings aan te

bring sal groot besparings aan toetsing en herstelling oor die lang termyn teweeg gebring word.

Toetsing begin by die vervaardigingsproses aangesien 'n bord wat nie getoets is voor verspreiding na die gebruiker, of wat foutief deurglip na swak toetsing, uiteidelik sy pad terug sal vind na die vervaardiger. Terug by die vervaardiger sal die bord getoets moet word, foute geïdentifiseer en die bord herstel word. Om bogenoemde probleme uit te skakel en die uiteindelike koste van die vervaardigde artikel so laag as moontlik te hou moet outomatiese toetsing gebruik word. Ten spyte van die ekstra tyd wat spandeer word om 'n makliker toetsbare ontwerp te verseker, sowel as die finansiering van addisionele komponente as gevolg van so 'n ontwerp, verseker ontwerp-om-te-kan-toets tegnieke dat totale kostes in terme van vervaardiging, toetsing en herstelling laag gehou word.

Die ontwikkelde metode kan vir KSI en MSI logika kringe gebruik word. Indien GSI (LSI) of BGSI (VLSI) logika betrokke is moet die strukturele aftas tegnieke oorweeg word. Aangesien baie tyd spandeer word om 'n toetsstel vir 'n bord saam te stel is outomatiese toetsing net lonend indien groot volumes van die spesifieke bord vervaardig gaan word.

In hierdie studie het dit duidelik geword dat outomatiese toetsing - waar sommige of al die besluitneming outomaties deur die rekenaar geneem word, met geen of weinige interaksie van die operateur - wel moontlik is. Die wenslikheid van opvolgstudies ten opsigte van strukturele tegnieke vir die opstel van toetsstelle het ook duidelik na vore gekom.

BYLAAG A

HOOFPROGRAM

```
Program Atsmain;


{**
Die hoofprogram word gebruik vir die opstelling van die menu.
Die sleutelbord word gelees om vas te stel watter sub-menu
uitgevoer moet word en of die program verlaat moet word.
Indien die program verlaat word, word  die skermkleure en die
wyser (cursor) herstel.
**}


Uses Crt, Dos, MenuUnit, AtsUnit;
Label GoOn;

Begin
     Setup;
GoOn:
     Repeat
       WindowNumber := 0;
       CurrentLine := 0;
       ReadKeyBoardHdg;
     Until (Ch = #27);
     Window(1,1,80,25);
     WriteAt(2,23,TextCol,TextBG,'Verlaat ATS? (J/N)');
       Repeat
         Sleutel := Upcase(ReadKey);
         If Sleutel = 'N' Then
           Begin
                 Window(2,23,79,23);
                 ClrScr;
                 Window(1,1,80,25);
                 Goto GoOn;
           End;
         If Sleutel = 'J' Then
           Begin
                 TextColor(White);
                 TextBackGround(Black);
                 ClrScr;
                 CursorOn;
           End;
       Until ( (Sleutel = 'J') OR (Sleutel = 'N') );
End.
```

Eenheid vir die beheer van die menu

```
UNIT MenuUnit;

{**
Die menueenheid word gebruik vir funksies rondom die
opstelling en beheer van die menu.
**}

INTERFACE

Uses DOS,CRT;
{$V-}

Const
     MaxScreen = 5;
     MenuName  = 'ATSMENU.MNU';
     MainHdg   = 'AUTOMATIESE TOETSSTELSEL';
     ErrorHdg  = 'BOODSKAPPE';
     Hdg1      = 'TOETS UITVOER';
     Hdg2      = 'SKEP NUWE LêER';
     Hdg3      = 'VERANDER BESTAANDE LêER';
     Hdg1Col   = 8;
     Hdg2Col   = 28;
     Hdg3Col   = 49;

Type
    ScreenImage = Array[0..3999] Of Byte;
    Pointr      = ^Integer;
    LnPtr       = ^MnuLn;
    MnuLn       = Record
                    MLine     : String[85];
                    Next,Last : LnPtr;
                  End;

Var
    ScreenPtr                          : Array[1..MaxScreen] Of
                                         ScreenImage;
    ScreenAddress                      : Word;
    DataPtr, TopHeap                   : Pointr;
    FirstHdg, LastHdg, NewHdg,
    FirstLine, LastLine,
    NewLine, TopLine, BotLine          : LnPtr;
    ThisLine                           : MnuLn;
    WindowNumber                       : Byte;
    MenuNumber                         : String[3];
    TextLine                           : String[30];
    Heading                            : String;
    MenuFile                           : Text;
    MenuFound, EndOfMenu               : Boolean;
    CurrentHdg, HdgRow, HdgCol,
    HdgLines, HdgOffset,
    MenuLines, CurrentLine             : Integer;
    CharacterOffset, AttributeOffset   : Integer;
```

```
     Ch, Key, Sleutel, ScrnType          : Char;
     Row, Col, CCol, CRow                : Integer;
     WinX1,WinY1,WinX2,WinY2             : Byte;
     TextCol, TextBG, HdgColor,
     MenuColor, HdgNormal, MenuNormal : Integer;
     Isolate, SingleStep, Show,
     SDelete, NewFile                    : String;
     Poort1A, Poort1B, Poort1CU,
     Poort1CL, Poort2A, Poort2B,
     Poort2CU, Poort2CL                  : String;
     ControlWord1, ControlWord2          : Byte;


Procedure SetUp;
Procedure CursorOff;
Procedure CursorOn;
Procedure ReadKeyBoardHdg;
Procedure WriteAt(CursX, CursY, TextCol, TextBG : Byte; S :
String);
Procedure SaveScreen(Var CCol,CRow :Integer);
Procedure RestoreScreen(CCol,CRow : Integer);
Procedure DrawBox(X1,Y1,X2,Y2,TextCol,TextBG : Byte);
Procedure MenuHighLight;
Procedure UpDown;
Procedure CalculateWindow;

IMPLEMENTATION

Uses AtsUnit;

Procedure ReadKeyBoardMenu; Forward;

{**
Prosedure "ColorSettings" verskaf die nodige kleure aan die
menu nadat vasgestel is of die skerm monochroom of kleur is.
**}

Procedure ColorSettings;
Begin
     If ScreenAddress = $B800 Then
       Begin {do color settings}
            TextCol    := (White);
            TextBG     := (Blue);
            HdgColor   := $7E;
            MenuColor  := $74;
            HdgNormal  := $1F;
            MenuNormal := $1F;
            ScrnType   := 'C';
       End {do color settings}
     Else
       If ScreenAddress = $B000 Then
         Begin {do monochrome settings}
              TextCol    := (White);
              TextBG     := (Black);
              HdgColor   := $F0;
              MenuColor  := $F0;
              HdgNormal  := $0F;
              MenuNormal := $0F;
              ScrnType   := 'M';
         End; {do monochrome settings}
```

```
     TextColor(TextCol);
     TextBackGround(TextBG);
End;


Procedure WriteAt(CursX,CursY,TextCol,TextBG : Byte;  S :
String);
Begin
     GotoXY(CursX,CursY);
     TextColor(TextCol);
     TextBackGround(TextBG);
     Write(S);
End;

{**
Prosedure "CursorOff" verwyder die wyser (cursor) vanaf die
skerm.
**}

Procedure CursorOff;
Var
   Regs : Registers;
Begin
     With Regs Do
       Begin
            AH := $01;
            CH := $20;
            CL := $20;
       End;
     Intr($10,Regs);
End;

{**
Prosedure "CursorOn" vertoon die wyser op die skerm nadat
"CursorOff" dit van die skerm verwyder het.
**}

Procedure CursorOn;
Var
   Regs : Registers;
Begin
     With Regs Do
       Begin
            If ScrnType = 'M' Then
              Begin
                   AH := 1; CH := 11; CL := 12;
              End
            Else
              Begin
                   AH := 1; CH := 6; CL := 7;
              End;
            Intr($10,Regs);
       End;
End;

{**
Prosedure "SaveScreen" stoor skerminligting na die geheue
van die rekenaar sodat dit later weer herroep kan word.
'n Maksimum van vyf skerms kan gestoor word. Die posisie
van die wyser word ook gestoor.
```

```
**}

Procedure SaveScreen(Var CCol, CRow : Integer);
Begin
     If(Mem[0000:1040] AND 48) <> 48 Then
       ScreenAddress := $B800
     Else ScreenAddress := $B000;
     If WindowNumber >= MaxScreen Then
       WriteAt(2,23,TextCol,TextBG,'Kan nie nog ''n skerm
stoor nie!!')
     Else
       Begin
            Inc(WindowNumber);
            CCol := WhereX;
            CRow := WhereY;
            Move(Mem[ScreenAddress:0000],ScreenPtr
[WindowNumber],4000);
       End;
End;

{**
"RestoreScreen" herroep die betrokke skerm vanuit geheue
en vertoon dit na die skerm. Die betrokke wyserposisie
word ook herstel.
**}

Procedure RestoreScreen(CCol, CRow : Integer);
Var
   ScreenAddress : Word;
Begin
     If(Mem[0000:1040] AND 48) <> 48 Then
       ScreenAddress := $B800
     Else ScreenAddress := $B000;

Move(ScreenPtr[WindowNumber],Mem[ScreenAddress:0000],4000);
     GotoXY(CCol,CRow);
     Dec(WindowNumber);
End;

{**
Prosedure "CalculateWindow" bereken die X en Y koördinate van
die aktiewe venster.
**}

Procedure CalculateWindow;
Begin
     WinX1 := Lo(WindMin)+1;
     WinY1 := Hi(WindMin)+1;
     WinX2 := Lo(WindMax)+1;
     WinY2 := Hi(WindMax)+1;
End;

{**
Prosedure "DrawBox" word gebruik vir die teken van
menuvensters deur gebruik te maak van ASCII-kode karakters.
**}

Procedure DrawBox(X1, Y1, X2, Y2, TextCol, TextBG : Byte);
Const
     UpLeftCon   = '';
     UpRightCon  = '';
```

```
      BotLeftCon  = '';
      BotRightCon = '';
      Horizontal  = '';
      Vertical    = '';
Var
      A : Byte;
Begin
      Window(1,1,80,25);
      WriteAt(X1,Y1,TextCol,TextBG,UpLeftCon);
      WriteAt(X2,Y1,TextCol,TextBG,UpRightCon);
      WriteAt(X1,Y2,TextCol,TextBG,BotLeftCon);
      WriteAt(X2,Y2,TextCol,TextBG,BotRightCon);
      For A := (X1+1) To (X2-1) Do
        Begin
             WriteAt(A,Y1,TextCol,TextBG,Horizontal);
             WriteAt(A,Y2,TextCol,TextBG,Horizontal);
        End;
      For A := (Y1+1) To (Y2-1) Do
        Begin
             WriteAt(X1,A,TextCol,TextBG,Vertical);
             WriteAt(X2,A,TextCol,TextBG,Vertical);
        End;
End;

{**
Prosedure "GetFile" lees die betrokke menu inligting uit
die lêer "atsmenu.mnu" en vertoon die inligting in die
toepaslike venster. Hoof-menu indelings word ook in die
lêer gestoor.
**}

Procedure GetFile;
Var
   Count : Integer;
Begin {initialize}
      HdgLines  := 0;
      MenuLines := 0;
      MenuFound := False;
      EndOfMenu := False;
      FirstHdg  := Nil;
      LastHdg   := Nil;
      FirstLine := Nil;
      LastLine  := Nil;
      Assign(MenuFile,MenuName);
      {$I-} Reset(MenuFile); {$S+}
      If IOResult <> 0 Then
        Begin {file not found}
             WriteAt(2,23,TextCol,TextBG,'Menu file not found.
Abort.');
             Halt;
        End;  {file not found}
      {mark top of the heap}
      Mark(TopHeap);
      Repeat
        Begin {search for desired menu}
             ReadLn(MenuFile,TextLine);
             If ((MenuNumber[1] = TextLine[1]) AND
                 (MenuNumber[2] = TextLine[2]) AND
                 (MenuNumber[3] = TextLine[3])) Then
               Begin {found a desired menu line}
```

```
                    MenuFound := True;
                    If ((TextLine[4] = '*') AND (TextLine[5] =
'H')
                        AND (TextLine[6] = '*')) Then
                      Begin {heading line}
                          For Count  := 1 To 6 Do
Delete(TextLine,1,1);
                          New(NewHdg);
                          HdgLines := HdgLines+1;
                          NewHdg^.Last := Nil;
                          NewHdg^.MLine := TextLine;
                          If FirstHdg = Nil Then
                            FirstHdg := NewHdg
                          Else
                            LastHdg^.Next := NewHdg;
                          LastHdg := NewHdg;
                          LastHdg^.Next := Nil;
                      End  {heading line}
                    Else
                        Begin {menu line}
                            For Count := 1 To 3 Do
Delete(TextLine,1,1);
                            New(NewLine);
                            MenuLines := MenuLines+1;
                            NewLine^.MLine := TextLine;
                            NewLine^.Last := LastLine;
                            If FirstLine = Nil Then
                              FirstLine := NewLine
                            Else
                              LastLine^.Next := NewLine;
                            LastLine := NewLine;
                            LastLine^.Next := FirstLine;
                            FirstLine^.Last := LastLine;
                        End; {menu line}
              End  {found a desired menu line}
            Else
              If MenuFound Then
                EndOfMenu := True;
      End; {search for desired menu}
    Until EndOfMenu OR EOF(MenuFile);
    If NOT MenuFound Then
      Begin {menu not found}
            WriteAt(2,23,TextCol,TextBG,'Menu number not
found. Abort.');
            Halt;
      End; {menu not found}
    Close(MenuFile);
End; {get menu file}

{**
Prosedure "MainWindow" word gebruik vir die teken van die
hoof-menuskerm.
**}

Procedure MainWindow;
Begin
    ClrScr;
    DrawBox(1,1,80,3,TextCol,TextBG);
    DrawBox(1,4,80,21,TextCol,TextBG);
    DrawBox(1,22,80,24,TextCol,TextBG);
```

```
     WriteAt( ((80-Length(MainHdg)) DIV
2),1,TextCol,TextBG,MainHdg);
     WriteAt( ((80-Length(ErrorHdg)) DIV
2),22,TextCol,TextBG,ErrorHdg);
End;

{**
"WriteHeadings" skryf die hoof-menu opskrifte nadat die
hoof-menuskerm vertoon is.
**}

Procedure WriteHeadings;
Begin
     NewHdg := FirstHdg;
     ThisLine.Mline := NewHdg^.Mline;
     WriteAt(9,2,TextCol,TextBG,ThisLine.Mline);
     NewHdg := NewHdg^.Next;
     ThisLine.MLine := NewHdg^.MLine;
     WriteAt(29,2,TextCol,TextBG,ThisLine.MLine);
     NewHdg := NewHdg^.Next;
     ThisLine.MLine := NewHdg^.MLine;
     WriteAt(50,2,TextCol,TextBG,ThisLine.MLine);
End;

{**
"WriteMenuSelections" skryf die toepaslike sub-menu
opskrifte in die betrokke vensters.
**}

Procedure WriteMenuSelections;
Var
   Line, Col : Integer;
Begin
     NewLine := FirstLine;
     TopLine := FirstLine;
     For Line := 1 To (MenuLines) Do
       Begin {write menu lines}
            ThisLine.Mline := NewLine^.Mline;
            BotLine := NewLine;
            GotoXY(1,Line);
            Write(' ',ThisLine.MLine);
            NewLine := NewLine^.Next;
       End; {write menu lines}
End;

{**
Prosedure "HdgInc" beweeg die wyser af na die volgende
menuseleksie. Indien die laaste menuseleksie gekies was
word die eerste menuseleksie gekies.
**}

Procedure HdgInc;
Begin
     If Heading = Hdg1 Then
       Begin
            Heading := Hdg2;
            HdgOffset := Hdg2Col;
       End
     Else
       If Heading = Hdg2 Then
```

```
          Begin
               Heading := Hdg3;
               HdgOffset := Hdg3Col;
          End
        Else
          If Heading = Hdg3 Then
            Begin
                 Heading := Hdg1;
                 HdgOffset := Hdg1Col;
                 CurrentHdg := 1;
            End;
End;

{**
Prosedure "HdgDec" beweeg die wyser op na die volgende
menuseleksie. Indien die eerste menuseleksie gekies was
word die laaste menuseleksie gekies.
**}

Procedure HdgDec;
Begin
     If Heading = Hdg1 Then
       Begin
            Heading := Hdg3;
            HdgOffset := Hdg3Col;
            CurrentHdg := 3;
       End
     Else
       If Heading = Hdg3 Then
         Begin
              Heading := Hdg2;
              HdgOffset := Hdg2Col;
         End
       Else
         If Heading = Hdg2 Then
           Begin
                Heading := Hdg1;
                HdgOffset := Hdg1Col;
           End;
End;

{**
"HdgHighLight" verhelder die gekose hoof-menu opskrif.
**}

Procedure HdgHighLight;
Begin
     For HdgCol := HdgOffset To ( HdgOffset-1 +
Length(Heading) ) Do
       Begin
            CharacterOffset := ( (HdgRow-1)*160 ) +
(HdgCol*2);
            AttributeOffset := CharacterOffset + 1;
            Mem[ScreenAddress : AttributeOffset] := HdgColor;
       End;
End;

{**
"MenuHighLight" verhelder die gekose sub-menu opskrif.
**}
```

```
Procedure MenuHighLight;
Begin
     Row := WinY1 + CurrentLine;
     For Col := (WinX1-1) To (WinX2-1) Do
       Begin
            CharacterOffset := ( (Row-1)*160 ) + (Col*2);
            AttributeOffset := CharacterOffset + 1;
            Mem[ScreenAddress : AttributeOffset] :=
MenuColor;
       End;
End;

{**
Prosedure "DoHdg1" word uitgevoer wanneer die menu "TOETS
UITVOER" gekies word. Die venster word geteken, die op-
sies ingeskryf en die eerste opsie verhelder. Daarna word
die sleutelbord gelees vir instruksies vanaf die
gebruiker.
**}

Procedure DoHdg1;
Begin
     SaveScreen(CCol,CRow);
     DrawBox(5,6,23,11,TextCol,TextBG);
     Window(6,7,22,10);
     MenuNumber := '002';
     GetFile;
     WriteMenuSelections;
     CalculateWindow;
     MenuHighLight;
     ReadKeyBoardMenu;
     RestoreScreen(CCol,CRow);
End;

{**
Prosedure "DoHdg2" word uitgevoer wanneer die menu "SKEP
NUWE LêER" gekies word. Dieselfde prosedure as sub-menu 1
word daarna gevolg.
**}

Procedure DoHdg2;
Begin
     SaveScreen(CCol,CRow);
     DrawBox(23,6,49,11,TextCol,TextBG);
     Window(24,7,48,10);
     MenuNumber := '003';
     GetFile;
     WriteMenuSelections;
     CalculateWindow;
     MenuHighLight;
     ReadKeyBoardMenu;
     RestoreScreen(CCol,CRow);
End;

{**
Prosedure "DoHdg3" word uitgevoer wanneer die menu "VER-
ANDER BESTAANDE LêER" gekies word. Dieselfde prosedure
as sub-menu 1 word daarna gevolg.
**}
```

```
Procedure DoHdg3;
Begin
     SaveScreen(CCol,CRow);
     DrawBox(51,6,72,11,TextCol,TextBG);
     Window(52,7,71,10);
     MenuNumber := '004';
     GetFile;
     WriteMenuSelections;
     CalculateWindow;
     MenuHighLight;
     ReadKeyBoardMenu;
     RestoreScreen(CCol,CRow);
End;

{**
Prosedure "ReadKeyBoardHdg" beweeg die verhelderblokkie
links of regs tussen die hoof-menu opsies, volgens die
instruksies wat van die sleutelbord ontvang word. Volgens
gebruikerskeuse word die betrokke sub-menu vertoon.
**}

Procedure ReadKeyBoardHdg;
Label NoKey;
Begin {do horizontal menus}
     CurrentLine := 0;
       Repeat
         Ch := ReadKey;
         If ( (Ch = #0) AND (KeyPressed) ) Then
           Ch := ReadKey
         Else
           If (Ch <> #13) Then
             Goto NoKey;
         If ORD(Ch) = 77 Then
           Begin {move cursor to the right}
                For HdgCol := HdgOffset To ( HdgOffset-1 +
Length(Heading) ) Do
                  Begin {normal video}
                       CharacterOffset := ( (HdgRow-1)*160 )
+ (HdgCol*2);
                       AttributeOffset := CharacterOffset +
1;
                       Mem[ScreenAddress:AttributeOffset] :=
HdgNormal;
                  End; {normal video}
                CurrentHdg := CurrentHdg+1;
                HdgInc;
                HdgHighLight;
           End; {move the cursor to the right}
           If ORD(Ch) = 75 Then
             Begin {move cursor to the left}
                  For HdgCol := HdgOffset To ( HdgOffset-1 +
Length(Heading) ) Do
                    Begin {normal video}
                         CharacterOffset := ( (HdgRow-1)*160
) + (HdgCol*2);
                         AttributeOffset := CharacterOffset +
1;
                         Mem[ScreenAddress:AttributeOffset]
:= HdgNormal;
                    End; {normal video}
```

```
                    CurrentHdg := CurrentHdg - 1;
                    HdgDec;
                    HdgHighLight;
                End; {move cursor to the left}
              CurrentLine := 0;
              Case ORD(Ch) of
                13 : Case CurrentHdg of
                  1 : DoHdg1;
                  2 : DoHdg2;
                  3 : DoHdg3;
                End; {case currenthdg}
              End; {case ch}
NoKey: Until Ch = #27;
End; {do horizontal menus}

{**
Prosedure "UpDown" beweeg die verhelderblokkie op en af
tussen die sub-menu opsies, volgens die instruksies wat
van die sleutelbord ontvang word.
**}

Procedure UpDown;
Begin
     Key := Readkey;
       If ( (Key = #0) AND (KeyPressed) ) Then
         Key := ReadKey
       Else If (Key <> #13) Then Exit;
     If ORD(Key) = 72 Then
       Begin {move cursor up}
         Begin {menu normal video}
              Row := WinY1 + CurrentLine;
              For Col := (WinX1-1) to (WinX2-1) Do
                Begin
                     CharacterOffset := ( (Row-1)*160 ) +
(Col*2);
                     AttributeOffset := CharacterOffset + 1;
                     Mem[ScreenAddress : AttributeOffset] :=
MenuNormal;
                End;
         End; {menu normal video}
         CurrentLine := CurrentLine - 1;
         If CurrentLine < 0 Then
           CurrentLine := MenuLines-1;
         MenuHighLight;
       End; {move cursor up}
     If ORD(Key) = 80 Then
       Begin {move cursor down}
            Row := WinY1 + CurrentLine;
            For Col := (WinX1-1) To (WinX2-1) Do
              Begin {menu normal video}
                   CharacterOffset := ( (Row-1)*160 ) +
(Col*2);
                   AttributeOffset := CharacterOffset + 1;
                   Mem[ScreenAddress : AttributeOffset] :=
MenuNormal;
              End; {menu normal video}
            CurrentLine := CurrentLine + 1;
            If CurrentLine >= MenuLines Then
              CurrentLine := 0;
            MenuHighLight;
```

```
        End; {move cursor down}
End;

{**
"ReadKeyBoardMenu" voer die verskillende opsies in elke
sub-menu uit volgens gebruikerskeuse.
**}

Procedure ReadKeyBoardMenu;
Begin {do vertical menus}
     Repeat
       UpDown;
       Case ORD(Key) of
         13 : Case CurrentHdg of
           1 : case CurrentLine of
             0 : ChooseFile;
             1 : Print;
             2 : Options;
             3 : Test;
           End; {case currentline of Hdg1}
           2 : Case CurrentLine of
             0 : ChooseFile;
             1 : IdentifyIO;
             2 : AddRecords;
             3 : ViewRecords;
           End; {case currentline of Hdg2}
           3 : Case CurrentLine of
             0 : ChooseFile;
             1 : ModRecords;
             2 : RemoveRecords;
             3 : AddEmptyRecords;
           End; {case currentline of Hdg3}
         End; {case currenthdg}
       End; {case Key}
     Until Key = #27;
End; {do vertical menus}

{**
Prosedure "SetUp" doen die nodige opstelling van die
program. Kleurverstellings word gedoen nadat vasgestel is
of 'n kleur- of monochroomskerm aan die rekenaar verbind is.
Beginwaardes word aan sekere veranderlikes toegeken
waarna die hoof-menuskerm opgestel word. Die wyser (ver-
derblokkie) word op die eerste hoof-menu opsie geplaas.
**}

Procedure SetUp;

Begin
     {determaine color or monochrome card}
     If (Mem[0000:1040] AND 48) <> 48 Then
       ScreenAddress := $B800
     Else ScreenAddress := $B000;
     ColorSettings;
     ClrScr;
     CursorOff;

     Isolate    := ' JA';
     SingleStep := 'NEE';
     Show       := ' JA';
```

```
        SDelete     := 'NEE';

        Poort1A     := 'UITSET';
        Poort1B     := 'UITSET';
        Poort1CU    := 'UITSET';
        Poort1CL    := 'UITSET';
        Poort2A     := 'UITSET';
        Poort2B     := 'UITSET';
        Poort2CU    := 'UITSET';
        Poort2CL    := 'UITSET';

        NewFile := ' ';
        MainWindow;
        MenuNumber := '001';
        GetFile;
        WriteHeadings;
        Heading := Hdg1;
        HdgOffset := Hdg1Col;
        HdgRow := 2;
        CurrentHdg := 1;
        HdgHighLight;
End;

End. {unit}
```

BYLAAG C

Eenheid vir die uitvoering van die OTS funksies

```
UNIT ATSUnit;

{**
"ATSUnit" bevat al die gebruikersopsies van die stelsel soos
gelys onder "INTERFACE". Hierdie opsies word opgeroep deur
die eenheid "MenuUnit".
**}

INTERFACE

Procedure ChooseFile;
Procedure Print;
Procedure Options;
Procedure Test;
Procedure IdentifyIO;
Procedure AddRecords;
Procedure ViewRecords;
Procedure ModRecords;
Procedure RemoveRecords;
Procedure AddEmptyRecords;

IMPLEMENTATION

Uses Crt, Dos, Printer, MenuUnit;
Type
    DirFiles = Array[1..200] Of String[13];
    TestRecord = Record
                       TestNr : String[8];
                       Vektor : Integer;
                       Result : Integer;
                       Pass   : String[8];
                       Fail   : String[8];
                 End;
    ResultRecord = Record
                         Vektor : Integer;
                         Result : Integer;
                         TestOP : Integer;
                   End;
Var
   ErrorCode      : Integer;
   WorkFile       : File Of TestRecord;
   TempFile       : File Of TestRecord;
   ResultFile     : File Of ResultRecord;
   TestRec        : TestRecord;
   ResultRec      : ResultRecord;
   ProgramDir     : String;
   WorkFileDir    : DirStr;
   Dir3           : DirStr;
   Name3          : NameStr;
   Ext3           : ExtStr;

{***
```

```
Prosedure "WriteError" skryf foute, of enige ander
inligting wat vir die gebruiker van belang mag wees,
na die FOUTBOODSKAP-vertoonarea.
***}

Procedure WriteError(S : String);
Var X, Y : Byte;
Begin
     X := WhereX;
     Y := WhereY;
     TextColor(TextCol+Blink);
     Window(2,23,79,23);
     GotoXY(2,23);
     Write(S);
     Window(2,5,79,20);
     TextColor(TextCol);
     GoToXY(X,Y);
End;


{***
Prosedure "ClearError" word gebruik om enige boodskappe
in die FOUTBOODSKAP-vertoonarea uit te wis.
***}

Procedure ClearError;
Var
   X,Y : Byte;
Begin
     X := WhereX;
     Y := WhereY;
     Window(2,23,79,23);
     ClrScr;
     Window(2,5,79,20);
     GotoXY(X,Y);
End;


{***
Prosedure "SoundError" lewer 'n waarskuwingstoon indien
'n fout deur die gebruiker begaan word.
***}

Procedure SoundError;
Begin
     Sound(880);
     Delay(200);
     NoSound;
End;

{**
Prosedure "CheckForError" word gebruik om te toets vir foute
wat na inset/uitset instruksies mag voorkom.
**}

Procedure CheckForError;
Begin
     ErrorCode := IOResult;
     If ErrorCode = 0 Then Exit;
     Repeat
```

```
        If ErrorCode = 152 Then
          WriteError(' Skyfaandrywer nie gereed nie. Maak deur
toe en druk enige sleutel...')
        Else
          WriteError(' ''n Probleem is ondervind. Herstel en
druk enige sleutel...');
        SoundError;
        Ch := ReadKey;
        ClearError;
        {$I-} ChDir(WorkFileDir) {$I+};
        ErrorCode := IOResult;
      Until ErrorCode <> 152;
End;

{***
Prosedure "CheckWorkFile"  word uitgevoer om te verseker
dat 'n lêer gekies is, voordat enige bewerkings of
instruksies uitgevoer word.
***}

Procedure CheckWorkFile;
Begin
      Repeat
        If NewFile = ' ' Then
          Begin
               WriteError('ENTER om lêer te kies, enige
sleutel om te verlaat');
               If Readkey = #13 Then
                 Begin
                      ClearError;
                      ChooseFile;
                 End
               Else
                 Begin
                      RestoreScreen(CCol,CRow);
                      Exit;
                 End;
          End;
      Until NewFile <> ' ';
End;

Procedure CheckResultFile;
Begin
      FSplit(NewFile,Dir3,Name3,Ext3);
        If ((Ext3 = '.TSR') OR (EXT3 = '.tsr')) Then
          Begin
               WriteError(' Verandering van resultaatlêer nie
toegelaat nie!!!');
               SoundError;
               Delay(2000);
               ClearError;
               RestoreScreen(CCol,CRow);
               Exit;
          End;
End;


{***
Prosedure "ChooseFile" word gebruik om 'n lêer te kies.
```

```
Nadat 'n lêer ingesle                    geverivieer of dit wel
bestaan. Indien van die wildcard "*" gebruik gemaak word,
word 'n lys van lêers verskaf. Toetse word ook uitgeoefen om
te verseker of die pad, skyfaandrywer ens. wel bestaan. 'n
Nuwe lêer kan ook geskep word vanaf hierdie opsie.
***}

Procedure ChooseFile;

Label Again, Create, Start;

Var
    ResultRec                : ResultRecord;}
    FileNumber               : Word;
    Path0, Path1             : String;
    i, FileCount             : Integer;
    ScrnFileCount            : Integer;
    df                       : DirFiles;
    Dir0, Dir1, Dir2         : DirStr;
    Name0, Name1             : NameStr;
    Ext0, Ext1               : ExtStr;
    f                        : File;
    DirString                : String;
    Keys                     : Char;
    X,Y                      : Byte;


Procedure TestForFile;
Begin
     ErrorCode := 0;
     GetDir(0,DirString);
     FSplit(Path0, Dir0, Name0, Ext0);
     Dir1 := Dir0;
     Delete( Dir1,Length(Dir1),1 );
     {$I-} ChDir(Dir1); {$I+}
     ErrorCode := IOResult;
     ChDir(DirString);
     If ((ErrorCode = 152) OR (ErrorCode = 3)) Then Exit;
     Assign(f,Path0);
     {$I-} Reset(f); {$I+}
     ErrorCode := IOResult;
     If ErrorCode = 0 Then
     Close(f);
End;

Procedure DirList( MaskIn : String;
                   Var NameList : DirFiles;
                   Var FileCounter : Integer);
Var
   i              : Byte;
   Regs           : Registers;
   DTASeg, DTAOfs : Word;
   FileName       : String[20];

Begin
     FillChar(Regs,SizeOf(Regs),0);
     FileCounter := 0;
     Regs.AH := $2F;
     MsDos(Regs);
     With Regs Do
       Begin
```

```
                DTASeg := ES;
                DTAOfs := BX;
          End;
        FillChar(Regs,SizeOf(Regs),0);
        MaskIn := MaskIn + #0;
        With Regs Do
          Begin
                AH := $4E;
                DS := Seg(MaskIn);
                DX := Ofs(MaskIn) + 1;
                CL := $00;
          End;
        MsDos(Regs);
        If Regs.AL <> 0 Then Exit;
        i := 1;
          Repeat
            FileName[i] := Chr(Mem[DTASeg:DTAOfs+29+i]);
            i := i + 1;
          Until (FileName[i-1] < #32) Or (i > 12);
        FileName[0] := Chr(i-1);
        FileCounter := 1;
        NameList[FileCounter] := FileName;
          Repeat
            FillChar(Regs,SizeOf(Regs),0);
            With Regs Do
              Begin
                    AH := $4F;
                    CL := $00;
              End;
            MsDos(Regs);
            If Regs.AL = 0 Then
              Begin
                    i := 1;
                    Repeat
                      FileName[i] :=
Chr(Mem[DTASeg:DTAOfs+29+i]);
                      i := i+1;
                    Until (FileName[i-1] < #32) Or (i >12);
                    Inc(FileCounter,1);
                    FileName[0] := Chr(i-1);
                    NameList[FileCounter] := FileName;
              End;
          Until Regs.AL <> 0;
End; {dirlist}


Begin
     SaveScreen(CCol,CRow);
Start:
     Window(2,5,79,20);
     ClrScr;
     GetDir(0,ProgramDir);
     Repeat
       ErrorCode := 0;
         Repeat {until path <> ''}
           WriteLn('Sleutel naam van lêer tesame met pad
in.');
           WriteLn;
           Write('-> ');
           CursorOn;
```

```
            ReadLn(Path0);
            CursorOff;
            ClrScr;
            ClearError;
          Until Path0 <> '';
        WriteLn;
        FSplit(Path0,Dir0,Name0,Ext0);
        WorkFileDir := Dir0;
AGAIN:  ErrorCode := 0;
        DirList(Path0,df,FileCount);
        If FileCount = 0 Then
          Begin {no files where found}
               TestForFile;
               If ErrorCode = 152 Then
                 Begin {disk door open}
                      Repeat {until errorcode <> 152}
                        WriteError('  Skyfaandrywer nie gereed
nie. Maak deur toe en druk enige sleutel...');
                        SoundError;
                        Ch := ReadKey;
                        ClearError;
                        TestForFile;
                      Until ErrorCode <> 152;
                      Goto AGAIN;
                 End; {disk door open}
               If ErrorCode = 3 Then
                 Begin
                      WriteError('  Pad nie gevind nie ');
                      SoundError;
                 End;
               If ErrorCode = 2 Then
                 Begin {test for empty directory}
                      Name1 := '*';
                      Ext1  := '.*';
                      Path1 := Dir0 + Name1 + Ext1;
                      DirList(Path1,df,FileCount);
                      If FileCount = 0 Then
                        Begin
                             WriteError(' Indeks nie gevind
nie ');
                             ErrorCode := 0;
                        End
                      Else
                        Begin
                             WriteError(' Lêer nie gevind nie
');
                             SoundError;
                             FSplit(Path0,Dir1,Name1,Ext1);
                             If Name1 = '*' Then
                               Begin
                                    Delay(1500);
                                    RestoreScreen(CCol,CRow);
                                    CursorOff;
                                    Exit;
                               End
                             Else
                               WriteLn('Moet ',Name1+Ext1,'
geskep word? (J/N)');
                             Repeat
                               Ch := UpCase(ReadKey);
```

```
                                   If Ch = 'J' Then
                                     Begin
                                          Assign(WorkFile,Path0);
                                          Rewrite(WorkFile);
                                          WriteLn;
                                          WriteLn(' Werklêer =
',Path0, '  Druk ENTER...');
                                          ClearError;
                                          Repeat
                                                Ch := Readkey
                                          Until Ch = #13;
                                          Close(WorkFile);

RestoreScreen(CCol,CRow);
                                          CursorOff;
                                          Exit;
                                     End
                                   Else
                                     Begin

RestoreScreen(CCol,CRow);
                                          Exit;
                                     End;
                                Until (Ch = 'J') OR (Ch = 'N');
                           End;
                  End; {test for empty directory}
               If ErrorCode = 5 Then
                  Begin
                       WriteError(' Toegang tot lêer geweier');
                       SoundError;
                  End;
               If ErrorCode = 150 Then
                  Begin
                       WriteError(' Diskette is skryfbeskermd
');
                       SoundError;
                  End;
          End; {no files where found}
     Until ErrorCode = 0;

     {write file names}
     ClrScr;
     ScrnFileCount := 0;
     For i := 1 To FileCount Do
       Begin
            WriteLn(i, ' - ',df[i]);
            Inc(ScrnFileCount);
            If ScrnFileCount = 12 Then
              Begin {wait}
                   WriteError(' Druk enige sleutel vir
volgende lêers... ');
                   Ch := ReadKey;
                   ScrnFileCount := 0;
                   ClrScr;
                   ClearError;
              End; {wait}
       End;
     WriteLn;
     WriteLn( (FileCount+1),' - Skep nuwe lêer');
     WriteLn( (FileCount+2),' - Kry nuwe directory');
```

```
      WriteLn( (FileCount+3),' - Keer terug na menu');
      WriteLn;
      Write('-> ');
      Write;
      WriteError(' Sleutel verlangde nommer in... ');
      X := WhereX;
      Y := WhereY;
      Repeat
        CursorOn;
        {$I-} ReadLn(FileNumber); {$I+}
        ErrorCode := IOResult;
        ClearError;
        CursorOff;
        If ErrorCode = 106 Then
          Begin
               WriteError('  Ongeldige numeriese formaat');
               SoundError;
               FileNumber := FileCount+4;     {stel filecount
buite perke}
          End;
        GoToXY(X,Y);
        ClrEol;
      Until ( (FileNumber <= (FileCount+3)) AND (ErrorCode =
0));
      ClearError;

      If FileNumber <= FileCount Then
        Begin {assign file}
             NewFile := df[FileNumber];
             FSplit(NewFile,Dir1,Name1,Ext1);
             If ((Ext1 = '.TSD') OR (Ext1 = '.tsd')) Then
               Begin
                    Assign(WorkFile,NewFile);
                    WriteLn('Werklˆer = ',NewFile);
                    WriteError(' Druk ENTER... ');
               End
             Else
             If ((Ext1 = '.TSR') OR (Ext1 = '.tsr')) Then
               Begin
                    Assign(ResultFile,NewFile);
                    WriteLn('Resultaatlêer = ',NewFile);
                    WriteError( 'Druk ENTER... ');
               End
             Else
               Begin
                    WriteError('Nie geldige lêer nie. (gebruik
.tsd of .tsr) Druk enige sleutel... ');
                    SoundError;
                    NewFile := ' ';
                    Ch := Readkey;
                    RestoreScreen(CCol,CRow);
                    Exit;
               End;
             Repeat
               Keys := Readkey;
             Until Keys = #13;
             RestoreScreen(CCol,CRow);
             Exit;
        End; {assign file}
```

```
      If FileNumber=(FileCount+1) Then
      Begin {create new databasis}
        Repeat {until errorcode = 0}
          Repeat
             ClrScr;
             WriteLn('Naam van databasis?');
             WriteLn;
             Write('-> ');
             CursorOn;
             ReadLn(NewFile);
             CursorOff;
             ClearError;
             FSplit(NewFile,Dir1,Name1,Ext1);
             If ((Ext1 <> '.TSD') AND (Ext1 <> '.tsd')) Then
               Begin
                    WriteError('Gebruik .tsd vir ''n nuwe
databasis. ');
                    SoundError;
               End;
          Until ((Ext1 = '.TSD') OR (Ext1 = '.tsd'));
             {$I-} ChDir(WorkFileDir) {$I+};
             CheckForError;
             Assign(WorkFile,NewFile);
             {$I-} Reset(WorkFile); {$I+}
             ErrorCode := IOResult;
             If ErrorCode = 0 Then
               Begin
                    Close(WorkFile);
                    WriteError(' Lêer bestaan alreeds. Druk
enige sleutel...');
                    SoundError;
                    Ch := ReadKey;
                    ChDir(ProgramDir);
                    RestoreScreen(CCol,CRow);
                    Exit;
               End;
             If ErrorCode <> 0 Then
               Begin
                    {$I-} Rewrite(WorkFile); {$I+}
                    Close(WorkFile);
                    ErrorCode := IOResult;
                    ChDir(ProgramDir);
                    If ErrorCode = 0 Then
                      Begin
                           WriteLn;
                           WriteLn('Werklêer = ',NewFile,'
Druk ENTER');
                      End
                    Else
                      Begin
                           WriteError('  Lêer kan nie geskep
word nie');
                           SoundError;
                      End;
               End;
        Until ErrorCode = 0;
        Repeat
          Ch := ReadKey;
        Until Ch = #13;
      End; {create new databasis}
```

```
      If FileNumber = (FileCount+2) Then
        Goto Start
      Else
       Begin
            RestoreScreen(CCol,CRow);
            CursorOff;
       End;
End;

{**
Die "Print" prosedure word gebruik om 'n die werklêer of
resultaatlêer te print. Fouttoetsing word gedurende die
uitvoering van hierdie prosedure gedoen.
**}

Procedure Print;
Label Again, Again1;
Type
    Str255 = String;
Var
   OneLine     : String;
   Lines       : LongInt;
   PrintStatus : Byte;
   StatusByte  : Integer;
   RecNr       : Word;

Procedure FormFeed;
Begin
     Write( Lst,Chr(12) )
End;

Function PrinterStat(WhichPrinter : Integer) : Integer;
Var
   Regs : Registers;
Begin {get printer status}
     Regs.AH := 2;
     Regs.AL := 0;
     Regs.DX := WhichPrinter;
     Intr(23,Regs);
     PrinterStat := Regs.AH;
End; {get printer status}


Procedure DoError;
Begin
     ClrScr;
     If PrintStatus = 56 Then
       Begin
            WriteError(' Daar is nie ''n drukker gekonnekteer
nie');
            SoundError;
       End
     Else
       If PrintStatus = 200 Then
         Begin
              WriteError(' Papier is op');
              SoundError;
         End
       Else
         If PrintStatus = 16 Then
```

```
            Begin
                 WriteError(' Drukker nie op lyn nie');
                 SoundError;
            End
          Else
            If PrintStatus = 48 Then
              Begin
                   WriteError(' Drukker is nie aangeskakel
nie');
                   SoundError;
              End;
            WriteLn('Herstel die probleem en druk ENTER...');
     Repeat
            {wait}
            Until ReadKey = #13;
            ClearError;
            ClrScr;
            PrintStatus := 144;
End; {do error stuff}


Begin
     SaveScreen(CCol,CRow);
     Window(2,5,79,20);
     ClrScr;
     CheckWorkFile;
     If NewFile = ' ' Then Exit;
     FSplit(NewFile,Dir3,Name3,Ext3);
     If ((Ext3 = '.TSD') OR (Ext3 = '.tsd')) Then
       WriteLn('Werkle^r = ',NewFile)
     Else
       WriteLn('Resultaatl^er = ',NewFile);
     WriteLn;
     WriteLn('Moet ''n drukstuk van ',NewFile,' gemaak word?
(J/N)');
     WriteLn;
     WriteLn('Program geskryf om met Seikosha2400-drukker te
funksioneer. Indien ander drukker gebruik word kan moontlike
probleme ondervind word.');
     WriteLn;
     Write('-> ');
     Sleutel := UpCase(ReadKey);
     If Sleutel = 'J' Then
       Begin {maak drukset}
            Lines := 0;
            RecNr := 0;
            {$I-} ChDir(WorkFileDir) {$I+};
            CheckForError;
            If ((Ext3 = '.TSD') OR (Ext3 = '.tsd')) Then
              Begin
                   Assign(WorkFile,NewFile);
                   Reset(WorkFile);
                   While NOT EOF(WorkFile) Do
                     Begin
Again:                     PrintStatus := PrinterStat(0);
                             If PrintStatus = 144 Then
                               Begin
                                    Read(Workfile,TestRec);
                                    With TestRec Do
```

```
                                        Begin
                                             {$I-}
WriteLn(Lst,RecNr:3,TestNr:8,' VEKT: ',Vektor:4,' RES:
',Result:4,
                                             ' SLAAG: ',Pass:8,'
FAAL: ',Fail:8) {$I+};

                                             Delay(5);
                                             ErrorCode :=
IOResult;
                                             If ErrorCode <> 0
Then Goto Again;
                                             Lines := Lines + 1;
                                             Inc(RecNr);
                                             If Lines MOD 56 = 0
Then FormFeed;
                                        End;
                               End
                             Else
                               DoError;
                        End;
               End;
             If ((Ext3 = '.TSR') OR (Ext3 = '.tsr')) Then
               Begin
                    Assign(ResultFile,NewFile);
                    Reset(ResultFile);
                    While NOT EOF(ResultFile) Do
                      Begin
Again1:                    PrintStatus := PrinterStat(0);
                           If PrintStatus = 144 Then
                             Begin
                                  Read(ResultFile,ResultRec);
                                  With ResultRec Do
                                    Begin
                                         {$I-} WriteLn(Lst,'
Vektor: ',Vektor:4,
                                         ' Resultaat:
',Result:4,' Uitset: ',TestOP:4) {$I+};
                                         Delay(5);
                                         ErrorCode :=
IOResult;
                                         If ErrorCode <> 0
Then Goto Again;
                                         Lines := Lines + 1;
                                         If Lines MOD 56 = 0
Then FormFeed;
                                    End;
                             End
                           Else
                             DoError;
                      End;
               End;
             If ((Ext3 = '.TSD') OR (Ext3 = '.tsd')) Then
               Close(WorkFile)
             Else
               Close(ResultFile);
             ChDir(ProgramDir);
        End; {maak drukset}
      If (Mem[0000:1040] AND 48) <> 48 Then
        ScreenAddress := $B800
```

```
      Else ScreenAddress := $B000;
      RestoreScreen(CCol,CRow);
End;

{**
Hierdie prosedure verskaf die gebruikersopsies aan die
gebruiker. Die opsies word deur middel van 'n keuseblok
op die skerm vertoon vanwaar dit dan verander kan word.
Die verloop van die uitvoering van 'n toets - sowel as
ander funksies - sal afhang van hierdie opsies. Wanneer
die program 'n aanvang neem word die standaard opsies
gebruik soos weergegee in prosedure "SetUp".
**}

Procedure Options;
Var
   X : Integer;

Procedure InvertOps(Ops : String);
Begin
      If Ops = ' JA' Then
        Ops := 'NEE'
      Else
        Ops := ' JA';
      Case CurrentLine Of
        0 : Isolate := Ops;
        1 : SingleStep := Ops;
        2 : Show := Ops;
        3 : SDelete := Ops;
      End;
End;

Begin
      SaveScreen(CCol,CRow);
      X := CurrentLine;
      CurrentLine := 0;
      DrawBox(15,10,43,15,TextCol,TextBG);
      Window(16,11,42,14);
      ClrScr;
      Repeat
        Window(16,11,42,14);
        WriteLn(' Isoleer fout          ',Isolate);
        WriteLn(' Enkelstap deur toets  ',SingleStep);
        WriteLn(' Vertoon alle stappe   ',Show);
        Write  (' Resultaatl^er uitwis  ',SDelete);
        Window(38,11,42,14);
        CalculateWindow;
        MenuHighLight;
        UpDown;
        Case ORD(Key) Of
          13 : Case CurrentLine Of
            0 : InvertOps(Isolate);
            1 : InvertOps(SingleStep);
            2 : InvertOps(Show);
            3 : InvertOps(SDelete);
          End; {currentline}
        End; {key}
      Until Key = #27;
      Key := ' ';
      CurrentLine := X;
```

```
     Window(6,7,22,10);
     CalculateWindow;
     RestoreScreen(CCol,CRow);
End;

{**
Die "Test" prosedure is die hoofprosedure en word gebruik
vir die uitvoering van die toetsstel (soos opgestel deur
die gebruiker) op 'n kring, om enige defekte in die kring
te identifiseer. Volgens die inligting in die toetsstel
word die uitvoering van die program beheer.
**}

Procedure Test;
Label Clock;
Var
   OutPortNr, InPortNr, MaskWord,
   Results, FaultFree              : Array[1..8] of Integer;
   O_Index, I_Index                : Integer;
   MaxOutP, MaxInP                 : Integer;
   FaultFile, Faulty, FaultNr      : Boolean;
   TempRec                         : String[7];
   CLKAddress, CLKCount,
   CLKStart, CLKStop               : Integer;
   Choice                          : Char;
   X, Y                            : Integer;

Procedure FaultInfo;
Begin
     SoundError;
     WriteError(' Die opstelling van die lêer is foutief.
Druk enige sleutel...');
     Ch := ReadKey;
     ClearError;
     FaultFile := True;
     CursorOff;
     ChDir(ProgramDir);
     RestoreScreen(CCol,CRow);
End;

{**
Prosedure "FaultyFile" word gebruik om foute wat voorgekom
het in die opstelling van 'n lêer te identifiseer.
**}

Procedure FaultyFile;
Begin
     ErrorCode := IOResult;
     If ErrorCode = 0 Then
       Exit;
     If ErrorCode = 152 Then
       Repeat
             WriteError(' Skyfaandrywer nie gereed nie. Maak
deur toe en druk enige sleutel...');
             SoundError;
             Ch := ReadKey;
             ClearError;
             {$I-} Read(WorkFile,TestRec) {$I+};
             ErrorCode := IOResult;
       Until ErrorCode <> 152
```

```
      Else
        FaultInfo;
End;

{**
Prosedure "ReadFaultInfo" lees die foutinligting vanuit die
lêer, nadat foutdiagnosering afgehandel is.
**}

Procedure ReadFaultInfo;
Var
   i : Integer;
Begin
      FaultNr := True;
      Repeat
        {$I-} Read(WorkFile,TestRec) {$I+};
        FaultyFile;
        If FaultFile = True Then Exit;
        If TestRec.TestNr = TempRec Then
          Begin
               FaultNr := False;
               WriteLn;
               WriteLn('Die volgende node/s is moontlik
foutief : ');
               WriteLn;
               For i := 1 To TestRec.Vektor Do
                 Begin
                      {$I-} Read(WorkFile,TestRec) {$I+};
                      FaultyFile;
                             If FaultFile = True Then Exit;
                      If ErrorCode <> 0 Then Exit;
                      WriteLn(TestRec.TestNr:9,
TestRec.Pass:9, TestRec.Fail:9);
                 End;
               ClearError;
          End;
        Until ((TestRec.TestNr = TempRec) OR (FaultNr =
False));
        Y := Y+TestRec.Vektor+5;
End; {get fault info}

{**
Prosedure "ReadVectorJump" neem die program na die volgende
vektor wat uitgevoer moet word.
**}

Procedure ReadVectorJump;
Begin {get vector jump}
      Repeat
        {$I-} Read(WorkFile,TestRec) {$I+};
        FaultyFile;
               If FaultFile = True Then Exit;
      Until (TestRec.TestNr = TempRec);
End; {get vector jump}


Begin
      SaveScreen(CCol,CRow);
      Window(2,5,79,20);
      ClrScr;
```

```
      CheckWorkFile;
      If NewFile = ' ' Then Exit;
      CheckResultFile;
      If ((Ext3 = '.TSR') OR (EXT3 = '.tsr')) Then Exit;
      X := 1;
      Y := 1;
      O_Index := 1;
      I_Index := 1;
      MaxOutP := 1;
      MaxInP  := 1;
      Faulty := False;
      FaultNr := False;
      FaultFile := False;
      {$I-} ChDir(WorkFileDir) {$I+};
      CheckForError;
      Assign(ResultFile,'Result.tsr');
      Rewrite(ResultFile);
      Reset(WorkFile);
      {$I-} Read(WorkFile,TestRec) {$I+};  {poort info}
      FaultyFile;
        If FaultFile = True Then Exit;
      If TestRec.TestNr <> 'ADDR' Then
        Begin
             FaultInfo;
             Exit;
        End;
      ControlWord1 := TestRec.Vektor;
      ControlWord2 := TestRec.Result;
      Port[$1B3] := ControlWord1;
      Port[$1B7] := ControlWord2;
      {$I-} Read(WorkFile,TestRec) {$I+};
      FaultyFile;
        If FaultFile = True Then Exit;
      Repeat
        If SingleStep = ' JA' Then
          Begin {wait for key to be pressed}
               WriteError(' Druk "N" om volgende vektor uit te
stuur');
               Repeat
                 Choice := Upcase(ReadKey);
               Until Choice = 'N';
               ClearError;
          End; {wait for key to be pressed}
        If FaultNr = True Then
          Begin
               {$I-} Read(WorkFile,TestRec) {$I+};
               FaultyFile;
                      If FaultFile = True Then Exit;
          End;

Clock: If TestRec.TestNr = 'CLK' Then
       Repeat
         Begin
               CLKAddress := TestRec.Vektor;
               CLKStart := TestRec.Result;
               {$I-} Read(WorkFile,TestRec) {$I+};
               FaultyFile;
               CLKCount := TestRec.Vektor;
               CLKStop := TestRec.Result;
               Repeat
```

```
                Port[CLKAddress] := CLKStart;
                Port[CLKAddress] := CLKStop;
                Dec(CLKCount);
              Until CLKCount = 0;
              {$I-} Read(WorkFile,TestRec) {$I+};
              FaultyFile;
                    If FaultFile = True Then Exit;
          End;
        Until TestRec.TestNr <> 'CLK';

        If TestRec.TestNr = 'CTRL' Then
          Repeat
            Port[TestRec.Vektor] := TestRec.Result;
            {$I-} Read(WorkFile,TestRec) {$I+};
            FaultyFile;
                  If FaultFile = True Then Exit;
          Until TestRec.TestNr <> 'CTRL';

        If TestRec.TestNr = 'CLK' Then
          Goto Clock;
        If TestRec.TestNr = 'O_PRT' Then
          Repeat
            OutPortNr[O_Index] := TestRec.Vektor;
            Inc(MaxOutP);
            Inc(O_Index);
            {$I-} Read(WorkFile,TestRec) {$I+};
            FaultyFile;
                  If FaultFile = True Then Exit;
          Until TestRec.TestNr <> 'O_PRT';

        If TestRec.TestNr = 'I_PRT' Then
          Repeat
            InPortNr[I_Index] := TestRec.Vektor;
            MaskWord[I_Index] := TestRec.Result;
            Inc(MaxInP);
            Inc(I_Index);
            {$I-} Read(WorkFile,TestRec) {$I+};
            FaultyFile;
                  If FaultFile = True Then Exit;
          Until TestRec.TestNr <> 'I_PRT';
          If TestRec.TestNr = 'CLK' Then
            Goto Clock;
        O_Index := 1;

        Repeat {skryf uit na poort}
          Port[OutPortNr[O_Index]] := TestRec.Vektor;
          ResultRec.Vektor := TestRec.Vektor;
          GotoXY(X,Y);
          If Show = ' JA' Then
            Begin
                 If O_Index < (MaxOutP-1) Then
                   Begin
                        WriteLn(' ADRES :
',OutPortNr[O_Index],'  UITSET na EOT : ',TestRec.Vektor:5);
                        ResultRec.Result := 0;
                        ResultRec.TestOP := 0;
                        Write(ResultFile,ResultRec);
                   End
                 Else
                   Begin
```

```
                        Write(' ADRES : ',OutPortNr[O_Index],'
UITSET na EOT : ',TestRec.Vektor:5);

                    End;
            End;
          Inc(O_Index);
          If O_Index < MaxOutP Then
            Begin
                {$I-} Read(WorkFile,TestRec); {$I+}
                FaultyFile;
                        If FaultFile = True Then Exit;
            End;
          If O_Index > MaxOutP Then
            Begin
                FaultInfo;
                Exit;
            End;
        Until O_Index = MaxOutP;

        I_Index := 1;
          Repeat {lees vanaf poort}
            FaultFree[I_Index] := TestRec.Result;
            ResultRec.TestOP := Port[InPortNr[I_Index]];
            ResultRec.TestOP := (ResultRec.TestOP AND
MaskWord[I_Index]);
            Results[I_Index] := ResultRec.TestOP;
            If Show = ' JA' Then
              WriteLn(' ADRES : ',InPortNr[I_Index],'  UITSET
gelees: ',ResultRec.TestOP)
            Else
              Write(' *');
            X := WhereX;
            Y := WhereY;
            ResultRec.Result := TestRec.Result;
            Write(ResultFile,ResultRec);
            ResultRec.Vektor := 0;
            Inc(I_Index);
            If I_Index < MaxInP Then
              Begin
                  {$I-} Read(WorkFile,TestRec); {$I+}
                  FaultyFile;
                        If FaultFile = True Then Exit;
              End;
          Until I_Index >= MaxInP;
          I_Index := 1;

          If Isolate = ' JA' Then
            Begin {isolate fault}
                Repeat
                  If Results[I_Index] <> FaultFree[I_Index]
Then
                    Faulty := True;
                  Inc(I_Index);
                Until ((I_Index = MaxInP) OR (Faulty =
True));
                If Faulty = False Then
                  Begin {resultaat korrek}
                      TempRec := TestRec.Pass;

Delete(TestRec.Pass,2,(Length(TestRec.Pass)-1));
```

```
                        If TestRec.Pass = 'f' Then
                          Begin {fout inligting}
                               ReadFaultInfo;
                                    If FaultFile = True Then
Exit;
                               If TempRec = 'f0' Then
                                 Begin
                                    FaultNr := False;
                                    Repeat
                                      {$I-}
Read(WorkFile,TestRec) {$I+};
                                      FaultyFile;
                                           If FaultFile =
True Then Exit;
                                    Until TestRec.TestNr =
'END';
                                    {$I-}
Read(WorkFile,TestRec) {$I+};
                                    FaultyFile;
                                         If FaultFile =
True Then Exit;
                                    MaxOutP := 1;
                                    MaxInP := 1;
                                    O_Index := 1;
                                    I_Index := 1;
                                 End
                               Else
                                 FaultNr := True;
                          End {fout inligting}
                        Else
                          Begin {vektor inligting}
                               ReadVectorJump;
                          End; {vektor inligting}
                  End {resultaat korrek}
                Else
                  Begin {resultaat foutief}
                       Faulty := False;
                       TempRec := TestRec.Fail;

Delete(TestRec.Fail,2,(Length(TestRec.Fail)-1));
                       If TestRec.Fail = 'f' Then
                         Begin {fout inligting}
                              ReadFaultInfo;
                              FaultNr := True;
                         End {fout inligting}
                       Else
                         ReadVectorJump;
                           If FaultFile = True Then Exit;
                  End; {resultaat foutief}
            End {isolate fault}
          Else
            Begin {go/no go}
                 Repeat
                   If Results[I_Index] = FaultFree[I_Index]
Then
                     Begin
                          {$I-} Read(WorkFile,TestRec) {$I+};
                          FaultyFile;
                                 If FaultFile = True Then
Exit;
```

```
                        If TestRec.TestNr = 'END' Then
                          WriteError('Kaart toets reg. Druk
ENTER');
                        Repeat
                          Ch := ReadKey;
                        Until Ch = #13;
                        ClearError;
                        Close(ResultFile);
                        Close(WorkFile);
                        ChDir(ProgramDir);
                        RestoreScreen(CCol,CRow);
                        Exit;
                    End
                  Else
                    Begin
                        WriteError('Kaart toets foutief.
Druk ENTER');
                        Repeat
                            Ch := ReadKey;
                        Until Ch = #13;
                        ClearError;
                        Close(ResultFile);
                        Close(WorkFile);
                        ChDir(ProgramDir);
                        RestoreScreen(CCol,CRow);
                        Exit;
                    End;
                Until ((I_Index = MaxInP) OR (Faulty =
True));
          End; {go/no go}
     Until ((TestRec.TestNr = 'END') OR (Faulty = True) OR
(FaultNr = True)) ;
     Close(ResultFile);
     Close(WorkFile);
     If SDelete = ' JA' Then
       Erase(ResultFile);
     ChDir(ProgramDir);
     WriteError(' Druk Enter...');
     Repeat
          Ch := ReadKey
     Until Ch = #13;
     ClearError;
     RestoreScreen(CCol,CRow);
End;

{**
Prosedure "IdentifyIO" verskaf 'n keuseblok vanwaar die
gebruiker die insette/uitsette op die twee 8255 PRKs
kan kies. Volgens die opstelling van die gebruiker
word die twee beheerwoorde bereken en as rekordnommer 0in die
werklêer gevoeg.
**}

Procedure IdentifyIO;
Var
   X, Y : Integer;

Procedure InvertPort(Poort : String);
Begin
     If Poort = ' INSET' Then
```

```
        Poort := 'UITSET
      Else
        Poort := ' INSET';
      Case CurrentLine Of
        0 : Poort1A  := Poort;
        1 : Poort1B  := Poort;
        2 : Poort1CU := Poort;
        3 : Poort1CL := Poort;
        4 : Poort2A  := Poort;
        5 : Poort2B  := Poort;
        6 : Poort2CU := Poort;
        7 : Poort2CL := Poort;
      End;
End;

Procedure CtrlWord1;
Var
   D7,D6,D5,D4,D3,D2,D1,D0 : Byte;
Begin
     D7 := 128;
     D6 := 0;
     D5 := 0;
     If Poort1A = ' INSET' Then
       D4 := 16
     Else
       D4 := 0;
     If Poort1CU = ' INSET' Then
       D3 := 8
     Else
       D3 := 0;
     D2 := 0;
     If Poort1B = ' INSET' Then
       D1 := 2
     Else
       D1 := 0;
     If Poort1CL = ' INSET' Then
       D0 := 1
     Else
       D0 := 0;
     ControlWord1 := D7+D6+D5+D4+D3+D2+D1+D0;
End;

Procedure CtrlWord2;
Var
   D7,D6,D5,D4,D3,D2,D1,D0 : Byte;
Begin
     D7 := 128;
     D6 := 0;
     D5 := 0;
     If Poort2A = ' INSET' Then
       D4 := 16
     Else
       D4 := 0;
     If Poort2CU = ' INSET' Then
       D3 := 8
     Else
       D3 := 0;
     D2 := 0;
     If Poort2B = ' INSET' Then
       D1 := 2
```

```
      Else
        D1 := 0;
      If Poort2CL = ' INSET' Then
        D0 := 1
      Else
        D0 := 0;
      ControlWord2 := D7+D6+D5+D4+D3+D2+D1+D0;
End;


Begin
      SaveScreen(CCol,CRow);
      CheckWorkFile;
      If NewFile = ' ' Then Exit;
      CheckResultFile;
      If ((Ext3 = '.TSR') OR (EXT3 = '.tsr')) Then Exit;
      X := Currentline;
      Y := MenuLines;
      Currentline := 0;
      MenuLines := 8;
      DrawBox(43,9,72,18,TextCol,TextBG);
      Window(44,10,71,17);
      ClrScr;
      Repeat
        WriteError(' Verskaf informasie m.b.v ENTER. Druk ESC
indien voltooi... ');
        Window(44,10,71,17);
        WriteLn('KANAAL 1 : POORT A   ' ,Poort1A);
        WriteLn('           POORT B   ' ,Poort1B);
        WriteLn('           POORT CH  ',Poort1CU);
        WriteLn('           POORT CL  ',Poort1CL);
        WriteLn('KANAAL 2 : POORT A   ' ,Poort2A);
        WriteLn('           POORT B   ' ,Poort2B);
        WriteLn('           POORT CH  ',Poort2CU);
        Write  ('           POORT CL  ',Poort2CL);
        Window(64,10,71,17);
        CalculateWindow;
        MenuHighLight;
        UpDown;
        Case ORD(key) Of
          13 : Case CurrentLine Of
            0 : InvertPort(Poort1A);
            1 : InvertPort(Poort1B);
            2 : InvertPort(Poort1CU);
            3 : InvertPort(Poort1CL);
            4 : InvertPort(Poort2A);
            5 : InvertPort(Poort2B);
            6 : InvertPort(Poort2CU);
            7 : InvertPort(Poort2CL);
          End; {currentline}
        End; {key}
      Until Key = #27;
      ClearError;
      WriteError(' Moet bogenoemde inligting na l^er geskryf
word? (J/N) ');
      Repeat
        Key := UpCase(Readkey);
        If Key = 'J' Then
          Begin
               CtrlWord1;
```

```
                CtrlWord2;
                {$I-} ChDir(WorkFileDir) {$I+};
                CheckForError;
                Assign(WorkFile,NewFile);
                Reset(WorkFile) ;
                TestRec.TestNr := 'ADDR';
                TestRec.Vektor := ControlWord1;
                TestRec.Result := ControlWord2;
                TestRec.Pass := '';
                TestRec.Fail := '';
                Write(WorkFile,TestRec);
                Close(WorkFile);
                ChDir(ProgramDir);
            End;
        Until (Key = 'J') OR (Key = 'N');
        GotoXY(5,5);
        Key := ' ';
        CurrentLine := X;
        MenuLines := Y;
        Window(24,7,48,9);
        CalculateWindow;
        RestoreScreen(CCol,CRow);
End;

{**
"CheckData" word gebruik om die geldigheid van data wat
deur die gebruiker in 'n lêer ingevoer word te bepaal.
**}

Procedure CheckData;
Begin
     ErrorCode := IOResult;
     If ErrorCode = 106 Then
       Begin
            WriteError(' Ongeldige numeriese formaat ');
            SoundError;
       End;
End;

{**
Prosedure "GetInputData" verkry vektorinligting wat as 'n
rekord gestoor moet word.
**}

Procedure GetInputData;

Begin
     ClrScr;
     CursorOn;
     Write('Sleutel Toetsnommer in (xxTxx) : ');
     ReadLn(TestRec.TestNr);
     Repeat
       Write('Sleutel toetsvektor in (desimaal) : ');
       {$I-} ReadLn(TestRec.Vektor) {$I+};
       ClearError;
       CheckData;
     Until ErrorCode = 0;
     Repeat
       Write('Sleutel foutvrye uitset in (desimaal) : ');
       {$I-} ReadLn(TestRec.Result) {$I+};
```

```
        ClearError;
        CheckData;
      Until ErrorCode = 0;
      Write('Sleutel bestemmingsvektor in wanneer foutvry
(desimaal) : ');
      ReadLn(TestRec.Pass);
      Write('Sleutel bestemmingsvektor in wanner faal
(desimaal) : ');
      ReadLn(TestRec.Fail);
      ClrScr;
      CursorOff;
End;

{**
Prosedure "AddRecords" verkry rekordinligting naamlik
fout-(FaultInfo), klok-(Clock), poort-(AddPort),
beheer-(ControlPort) en vektorinligting (AddRecords)
en voeg dit by die werklêer. 'n "END" kan ook 'n die
werklêer gevoeg word deur middel van prosedure "Ending"
Prosedure "GetInputData" vir vektorinligting word
buite hierdie prosedure gegee aangesien dit ook by
"verander rekord" gebruik word.
**}

Procedure AddRecords;
Var
   Choice    : Char;
   LenOfPage : Byte;
   i         : Word;

Procedure AddVector;
Begin
      GetInputData;
      {$I-} ChDir(WorkFileDir) {$I+};
      CheckForError;
      Reset(WorkFile);
      Seek(WorkFile,FileSize(WorkFile));
      Write(WorkFile,TestRec);
      Close(WorkFile);
      ChDir(ProgramDir);
End;


Procedure AddPort;
Begin
      CursorOn;
      WriteLn('Moet inset/uitset poort bygevoeg word? (I/U)');
      Repeat
        Choice := Upcase(ReadKey);
        If Choice = 'U' Then
          Begin {uitset}
               Repeat
                 Write('Watter poort moet as uitset gebruik
word? -> ');
                 {$I-} ReadLn(TestRec.Vektor) {$I+};
                 ClearError;
                 CheckData;
               Until ErrorCode = 0;
               TestRec.TestNr := 'O_PRT';
               TestRec.Result := 0;
```

```
          End; {uitset}
        If Choice = 'I' Then
          Begin {inset}
               Repeat
                 Write('Watter poort moet as inset gebruik
word? -> ');
                 {$I-} ReadLn(TestRec.Vektor) {$I+};
                 ClearError;
                 CheckData;
               Until ErrorCode = 0;
               Repeat
                 Write('Sleutel maskergreep in -> ');
                 {$I-} ReadLn(TestRec.Result) {$I+};
                 ClearError;
                 CheckData;
               Until errorCode = 0;
               TestRec.TestNr := 'I_PRT';
          End; {inset}
     Until ((Choice = 'U') OR (Choice = 'I'));
     CursorOff;
     TestRec.Pass := '';
     TestRec.Fail := '';
     {$I-} ChDir(WorkFileDir) {$I+};;
     CheckForError;
     Reset(WorkFile);
     Seek(WorkFile,FileSize(WorkFile));
     Write(WorkFile,TestRec);
     Close(WorkFile);
     ChDir(ProgramDir);
End;

Procedure ControlPort;
Begin
     CursorOn;
     Repeat
       Write('Watter poort moet ''n beheer poort gemaak word?
-> ');
       {$I-} ReadLn(TestRec.Vektor) {$I+};
       ClearError;
       CheckData;
     Until ErrorCode = 0;
     Repeat
       Write('Watter beheerwoord moet gebruik word? -> ');
       {$I-} ReadLn(TestRec.Result) {$I+};
       ClearError;
       CheckData;
     Until ErrorCode = 0;
     CursorOff;
     TestRec.TestNr := 'CTRL';
     TestRec.Pass := '';
     TestRec.Fail := '';
     {$I-} ChDir(WorkFileDir) {$I+};
     CheckForError;
     Reset(WorkFile);
     Seek(WorkFile,FileSize(WorkFile));
     Write(WorkFile,TestRec);
     Close(WorkFile);
     ChDir(ProgramDir);
End;
```

```
Procedure FaultInfo;
Var
   i : Integer;
Begin
     CursorOn;
     Write('Wat is die foutnommer? -> ');
     ReadLn(TestRec.TestNr);
     Repeat
       Write('Hoeveel rekords wil gebruik word vir nodes? ->
');
       {$I-} ReadLn(TestRec.Vektor) {$I+};
       ClearError;
       CheckData;
     Until ErrorCode = 0;
     CursorOff;
     TestRec.Result := 0;
     TestRec.Pass := '';
     TestRec.Fail := '';
     {$I-} ChDir(WorkFileDir) {$I+};;
     CheckForError;
     Reset(WorkFile);
     Seek(WorkFile,FileSize(WorkFile));
     Write(WorkFile,TestRec);
     For i := 1 To TestRec.Vektor Do
       Begin
            CursorOn;
            Write('Foutiewe node inligting? -> ');
            ReadLn(TestRec.TestNr);
            Write('Foutiewe node inligting? -> ');
            ReadLn(TestRec.Pass);
            Write('Foutiewe node inligting? -> ');
            ReadLn(TestRec.Fail);
            TestRec.Vektor := 0;
            TestRec.Result := 0;
            {$I-} ChDir(WorkFileDir) {$I+};
            CheckForError;
            Seek(WorkFile,FileSize(WorkFile)) ;
            Write(WorkFile,TestRec);
            CursorOff;
       End;
     Close(WorkFile);
     ChDir(ProgramDir);
End;

Procedure Clock;
Begin
     CursorOn;
     Repeat
       Write('Watter poort moet ''n klok gemaak word? -> ');
       {$I-} ReadLn(TestRec.Vektor) {$I+};
       ClearError;
       CheckData;
     Until ErrorCode = 0;
     Repeat
       Write('Wat is die begin pulswaarde? -> ');
       {$I-} ReadLn(TestRec.Result) {$I+};
       ClearError;
       CheckData;
     Until ErrorCode = 0;
     CursorOff;
```

```
      TestRec.TestNr := 'CLK';
      TestRec.Pass := '';
      TestRec.Fail := '';
      {$I-} ChDir(WorkFileDir) {$I+};
      CheckForError;
      Reset(WorkFile);
      Seek(WorkFile,FileSize(WorkFile));
      Write(WorkFile,TestRec);
      CursorOn;
      Repeat
        Write('Wat is die end pulswaarde? -> ');
        {$I-} ReadLn(TestRec.Result) {$I+};
        ClearError;
        CheckData;
      Until ErrorCode = 0;
      Repeat
        Write('Hoeveel pulse moet gestuur word? -> ');
        {$I-} ReadLn(TestRec.Vektor) {$I+};
        ClearError;
        CheckData;
      Until ErrorCode = 0;
      CursorOff;
      {$I-} ChDir(WorkFileDir) {$I+};
      CheckForError;
      Write(WorkFile,TestRec);
      Close(WorkFile);
      ChDir(ProgramDir);
End;

Procedure Ending;
Begin
      TestRec.TestNr := 'END';
      TestRec.Vektor := 0;
      TestRec.Result := 0;
      TestRec.Pass   := '';
      TestRec.Fail   := '';
      {$I-} ChDir(WorkFileDir) {$I+};;
      CheckForError;
      Reset(WorkFile);
      Seek(WorkFile,FileSize(WorkFile));
      Write(WorkFile,TestRec);
      Close(WorkFile);
      ChDir(ProgramDir);
End;


Begin
      SaveScreen(CCol,CRow);
      Window(2,5,79,20);
      ClrScr;
      CheckWorkFile;
      If NewFile = ' ' Then Exit;
      CheckResultFile;
      If ((Ext3 = '.TSR') OR (EXT3 = '.tsr')) Then Exit;
      {$I-} ChDir(WorkFileDir) {$I+};
      CheckForError;
      Reset(WorkFile);
      If (FileSize(WorkFile) = 0) Then
        Begin
             Close(WorkFile);
```

```
            WriteError(' Die inset/uitset poorte moet eers
identifiseer word');
            SoundError;
            Delay(2500);
            ChDir(ProgramDir);
            RestoreScreen(CCol,CRow);
            Exit;
        End;
      ChDir(ProgramDir);
      Repeat
        ClrScr;
        WriteLn('WerkLe^r = ',NewFile);
        WriteLn('A - voeg rekord by     P - voeg poort by    C
- voeg beheer poort');
        WriteLn('F - fout inligting     K - voeg klok by     E
- voeg END by');
        WriteLn('V - vertoon l^er       ESC - verlaat');
        If Choice = #0 Then
          Choice := Upcase(ReadKey);
        Choice := Upcase(ReadKey);
        Case Choice Of
          'A' : AddVector;
          'V' : ViewRecords;
          'P' : AddPort;
          'C' : ControlPort;
          'F' : FaultInfo;
          'K' : Clock;
          'E' : Ending;
        End; {choice}
      Until Choice = #27;
      RestoreScreen(CCol,CRow);
End;

{**
Prosedure "ViewRecords" vertoon die inhoud van die werklêer
of resultaatlêer na die skerm. Deur gebruik te maak van PgUp,
PgDn, Home en End kan die gebruiker die inhoud van die
betrokke lêer nagaan. Rekordnommers word saam met die inhoud
van die werklêer vertoon.
**}

Procedure ViewRecords;
Const
      Esc     = #27;
      HomeKey = #71;
      EndKey  = #79;
      PgUp    = #73;
      PgDn    = #81;
Var
   Choice  : Char;
   Ch      : Char;
   DoWrite : Boolean;
   RecNr   : Word;
   Dir     : DirStr;
   Name    : NameStr;
   Ext     : ExtStr;

Begin
      SaveScreen(CCol,CRow);
      Window(2,5,79,20);
```

```
      ClrScr;
      CheckWorkFile;
      If NewFile = ' ' Then Exit;
      RecNr := 0;
      {$I-} ChDir(WorkFileDir) {$I+};;
      CheckForError;
      FSplit(NewFile,Dir3,Name3,Ext3);
      If ((EXT3 = '.TSD') OR (EXT3 = '.tsd')) Then
      Begin
        Assign(WorkFile,NewFile);
        Reset(WorkFile);
        If FileSize(WorkFile) = 0 Then
          WriteError('L^er bevat geen inhoud nie. Druk
ESC...');
        While ( NOT EOF(WorkFile) OR (Choice <> Esc) )Do
          Begin
               DoWrite := True;
               If NOT EOF(WorkFile) Then
                 Read(WorkFile,TestRec)
               Else
                 DoWrite := False;
               With TestRec Do
               Begin
                    If DoWrite = True Then
                      WriteLn(RecNr:3,TestNr:8,' VEKTOR :
',Vektor:4,' RESULTAAT : ',Result:4,
                       ' SLAAG : ',Pass:8,' FAAL : ',Fail:8);
                    Inc(RecNr);
                    If ( (Filepos(WorkFile) MOD 14) = 0) OR
(DoWrite = False) Then
                    Repeat
                      If FileSize(WorkFile) >= 14 Then
                      WriteError(' Gebruik PgUp,PgDn,Home,End
of Esc');
                      Choice := ReadKey;
                      If Choice = #0 Then
                        Choice := ReadKey;
                      Case Choice Of
                        HomeKey : If (FilePos(WorkFile) < 15)
Then
                                    Begin
                                         SoundError;
                                         DoWrite := False;
                                         Choice := ' ';
                                    End
                                  Else
                                    Begin
                                         ClrScr;
                                         Seek(WorkFile,0);
                                         RecNr := 0;
                                         DoWrite := False;
                                    End;
                             EndKey : If NOT EOF(WorkFile) Then
                                       Begin
                                            ClrScr;
                                            If (FileSize(WorkFile)
MOD 14) = 0 Then
                                              Begin

Seek(WorkFile,FileSize(WorkFile)-14);
```

```
                                                  RecNr :=
(FileSize(WorkFile)-14);
                                              End
                                            Else
                                              Begin

Seek(WorkFile,FileSize(WorkFile)-(FileSize(WorkFile) MOD
14));
                                                  RecNr :=
FileSize(WorkFile)-(FileSize(WorkFile) MOD 14);
                                              End;
                                      End
                                    Else
                                      Begin
                                           WriteLn('<< Einde van
lêer >>');
                                           SoundError;
                                           Choice := ' ';
                                      End;
                          PgDn : If (RecNr >=
FileSize(WorkFile)) Then
                                    Begin
                                         WriteLn('<< Einde van
lêer >>');
                                         SoundError;
                                         Choice := ' ';
                                    End
                                  Else
                                    Begin
                                         ClrScr;
                                    End;
                          PgUp : If FilePos(WorkFile) < 15 Then
                                    Begin
                                         ClrScr;
                                         Seek(WorkFile,0);
                                         RecNr := 0;
                                         SoundError;
                                    End
                                  Else
                                    Begin
                                         ClrScr;
                                         If ((FilePos(WorkFile)
MOD 14) = 0) Then
                                           Begin

Seek(WorkFile,FilePos(WorkFile)-28);
                                                RecNr := RecNr -
28
                                           End
                                         Else
                                           Begin

Seek(WorkFile,FilePos(WorkFile)-(14+(FilePos(WorkFile) MOD
14) ));
                                                RecNr := RecNr-
(13+(RecNr MOD 13));
                                           End;
                                    End;
                          Esc : Begin
```

```
Seek(WorkFile,FileSize(WorkFile));
                                        DoWrite := False;
                                    End;
                              End; {case choice}
                            Until (Choice = Esc) OR (Choice = HomeKey)
OR (Choice = EndKey)
                             OR (Choice = PgUp) OR (Choice = PgDn);
                 End;
          End;
       Close(WorkFile);
       End;

    If ((EXT3 = '.TSR') OR (EXT3 = '.tsr')) Then
    Begin
      Assign(ResultFile,NewFile);
      Reset(ResultFile);
      If FileSize(ResultFile) = 0 Then
        WriteError('Lêer bevat geen inhoud nie. Druk ESC...');
      While ( NOT EOF(ResultFile) OR (Choice <> Esc) )Do
        Begin
             DoWrite := True;
             If NOT EOF(ResultFile) Then
               Read(ResultFile,ResultRec)
             Else
               DoWrite := False;
             With ResultRec Do
             Begin
                  If DoWrite = True Then
                    WriteLn('VEKTOR : ',Vektor:4,' FOUTVRYE
UITSET : ',Result:4,
                      ' UITSET GELEES: ', TestOP:4);
                  If ( (Filepos(ResultFile) MOD 14) = 0) OR
(DoWrite = False) Then
                  Repeat
                    If FileSize(ResultFile) >= 14 Then
                    WriteError(' Gebruik PgUp,PgDn,Home,End of
Esc');
                    Choice := ReadKey;
                    If Choice = #0 Then
                      Choice := ReadKey;
                    Case Choice Of
                      HomeKey : If (FilePos(ResultFile) < 15)
Then
                                  Begin
                                       SoundError;
                                       DoWrite := False;
                                       Choice := ' ';
                                  End
                                Else
                                  Begin
                                       ClrScr;
                                       Seek(ResultFile,0);
                                       DoWrite := False;
                                  End;
                      EndKey : If NOT EOF(ResultFile) Then
                                   Begin
                                       ClrScr;
                                       If (FileSize(ResultFile)
MOD 14) = 0 Then
Seek(ResultFile,FileSize(ResultFile)-14)
```

```
                                        Else

Seek(ResultFile,FileSize(ResultFile)-(FileSize(ResultFile)
MOD 14))
                                      End
                                    Else
                                      Begin
                                           WriteLn('<< Einde van
lêer >>');
                                           SoundError;
                                           Choice := ' ';
                                      End;
                          PgDn : If (FilePos(ResulTFile) >=
FileSize(ResultFile)) Then
                                    Begin
                                         WriteLn('<< Einde van lêer
>>');
                                         SoundError;
                                         Choice := ' ';
                                    End
                                  Else
                                    Begin
                                         ClrScr;
                                    End;

                         PgUp : If FilePos(ResultFile) < 15 Then
                                    Begin
                                         ClrScr;
                                         Seek(ResultFile,0);
                                         SoundError;
                                    End
                                  Else
                                    Begin
                                         ClrScr;
                                         If ((FilePos(ResultFile)
MOD 14) = 0) Then

Seek(ResultFile,FilePos(ResultFile)-28)
                                         Else

Seek(ResultFile,FilePos(ResultFile)-(14+(FilePos(ResultFile)
MOD 14) ));
                                    End;

                          Esc : Begin

Seek(ResultFile,FileSize(ResultFile));
                                      DoWrite := False;
                                End;
                      End; {case choice}
                    Until (Choice = Esc) OR (Choice = HomeKey)
OR (Choice = EndKey)
                       OR (Choice = PgUp) OR (Choice = PgDn);
               End;
         End;
       Close(ResultFile);
       End;
       ChDir(ProgramDir);
       RestoreScreen(CCol,CRow);
End;
```

```
{**
Prosedure "ModRecords" word gebruik om 'n rekord in 'n
bestaande lêer te verander. Dit word gedoen deur 'n
tydelike lêer te skep. Die inligting voor die rekordnommer
(wat verander moet word) word oorgeskryf na die tydelike
lêer gevolg deur die rekord wat verander moet word. Die
res van die lêer word nou oorgeskryf na die tydelike lêer
waarna die werklêer uitgewis word en die naam van die
tydelike lêer verander word na die werklêer benaming.
**}

Procedure ModRecords;
Var
   x    : Word;
   Ch   : Char;


Procedure Modify;
Var
   r : Word;
Begin
     ClrScr;
     {$I-} ChDir(WorkFileDir) {$I+};
     CheckForError;
     Assign(TempFile,'Temp.TSD');
     Rewrite(TempFile);
     Reset(WorkFile);
     With TestRec Do
       Begin
            If x <> 0 Then
            Begin
                 For r := 0 To (x-1) Do
                   Begin
                        Read(WorkFile,TestRec);
                        Write(TempFile,TestRec);
                   End;
            End;
            Close(WorkFile);
            Close(TempFile);
            GetInputData;
            Reset(TempFile);
            Seek(TempFile,x);
            Write(TempFile,TestRec);
            Reset(WorkFile);
            Seek(WorkFile,x+1);
            For r := (x+1) To (FileSize(WorkFile)-1) Do
              Begin
                   Read(WorkFile,TestRec);
                   Write(TempFile,TestRec);
              End;
            Close(WorkFile);
            Erase(Workfile);
            Close(TempFile);
            Rename(TempFile,NewFile);
            ChDir(ProgramDir);
     End; {with}
End;

Begin
     SaveScreen(CCol,Crow);
```

```
      Window(2,5,79,20);
      ClrScr;
      CheckWorkFile;
      If NewFile = ' ' Then Exit;
      CheckResultFile;
      If ((Ext3 = '.TSR') OR (EXT3 = '.tsr')) Then Exit;
      {$I-} ChDir(WorkFileDir) {$I+};;
      CheckForError;
      Reset(WorkFile);
      Repeat
        ClrScr;
        CursorOn;
        Repeat
          Write(' Rekordnommer om te verander? -> ');
          WriteError(' Sleutel "0" in om opsie te
verlaat...');
          {$I-} ReadLn(x) {$I+};
          ClearError;
          CheckData;
        Until ErrorCode = 0;
        CursorOff;
        If x = 0 Then
          Begin
               WriteError(' Gebruik "Identifiseer I/U poorte"
indien rekord 0 verander wil word.');
               SoundError;
               Delay(3500);
               ClearError;
               RestoreScreen(CCol,CRow);
               ChDir(ProgramDir);
               Exit;
          End;
        If x > (FileSize(WorkFile)-1) Then
          Begin
               WriteError(' Rekordnommer val buite bestek van
lêer');
               SoundError;
          End;
      Until x < FileSize(WorkFile);
      With TestRec Do
        Begin
            Seek(WorkFile,x);
            {$I-} ChDir(WorkFileDir) {$I+};
            CheckForError;
            Read(WorkFile,Testrec);
            WriteLn(x:3,TestNr:6,' VEKTOR : ',Vektor:4,'
RESULTAAT : ',Result:4,
            ' PASS : ',Pass:5,' FAIL : ',Fail:5);
            Close(WorkFile);
            ChDir(ProgramDir);
            WriteLn;
            CursorOn;
            Write(' Moet die rekord verander word? J/N');
            Repeat
              Ch := UpCase(ReadKey);
            Until (Ch = 'J') OR (Ch = 'N');
            Case Ch Of
              'N' : {exit};
              'J' : Modify
            End; {case ch}
```

```
            CursorOff;
        End;
     RestoreScreen(CCol,CRow);
End;

{**
Prosedure "RemoveRecords" word gebruik om enige aantal
rekords vanuit die werklêer te verwyder. Die gebruik van
'n tydelike lêer is, net soos by die verandering van 'n
lêer, ook hier toegepas.
**}

Procedure RemoveRecords;
Var
   x,y,r : Word;

Begin
     SaveScreen(CCol,Crow);
     Window(2,5,79,20);
     ClrScr;
     CheckWorkFile;
     If NewFile = ' ' Then Exit;
     CheckResultFile;
     If ((Ext3 = '.TSR') OR (EXT3 = '.tsr')) Then Exit;
     Repeat
       ClrScr;
       CursorOn;
       Repeat
         Write('Nommer van begin rekord om te verwyder? ->
');
         {$I-} ReadLn(x) {$I+};
         ClearError;
         CheckData;
       Until ErrorCode = 0;
       Repeat
         Write('Nommer van end rekord om te verwyder? -> ');
         {$I-} ReadLn(y) {$I+};
         ClearError;
         CheckData;
       Until ErrorCode = 0;
       If ((X = 0) OR (Y = 0)) Then
         Begin
              WriteError(' Gebruik "Identifiseer I/U poorte"
om rekord 0 te verander.');
              SoundError;
              Delay(2500);
              ClearError;
              CursorOff;
              RestoreScreen(CCol,CRow);
              ChDir(ProgramDir);
              Exit;
         End;
       CursorOff;
       If x > y Then
         Begin
              WriteError(' Kleiner waarde moet eerste
ingesleutel word');
              SoundError;
         End;
     Until y >= x;
```

```
     {$I-} ChDir(WorkFileDir) {$I+};
     CheckForError;
     Assign(TempFile,'Temp.TSD');
     Rewrite(TempFile);
     Reset(WorkFile);
     If y >= FileSize(WorkFile) Then
       Begin
            WriteError(' Rekordnommer/s wat aangevra is val
buite bestek van l^er...');
            SoundError;
            Close(WorkFile);
            Close(TempFile);
            Erase(TempFile);
            Delay(2500);
            RestoreScreen(CCol,CRow);
            ChDir(ProgramDir);
            Exit;
       End;
     With TestRec Do
       Begin
            If x = 0 Then
              Begin
                   Seek(WorkFile,y+1);
                   For r := (y+1) To (FileSize(WorkFile)-1)
Do
                     Begin
                          Read(WorkFile,TestRec);
                          Write(TempFile,TestRec);
                     End;
              End
            Else
              Begin
                   For r := 0 To (x-1) Do
                     Begin
                          Read(WorkFile,TestRec);
                          Write(TempFile,TestRec);
                     End;
                   Seek(WorkFile,y+1);
                   For r := (y+1) To (FileSize(WorkFile)-1)
Do
                     Begin
                          Read(WorkFile,TestRec);
                          Write(TempFile,TestRec);
                     End;
              End;
       End;
     Close(WorkFile);
     Erase(Workfile);
     Close(TempFile);
     Rename(TempFile,NewFile);
     ChDir(ProgramDir);
     RestoreScreen(CCol,CRow);
End;

{**
Prosedure "AddEmptyRecords" word gebruik om addisionele
rekords, volgens die rekordnommers, in die werklêer te
plaas. Bogenoemde tegniek word weereens gebruik vir die
uitvoering van hierdie prosedure.
**}
```

```
Procedure AddEmptyRecords;
Var
   x,r,s : Word;
Begin
     SaveScreen(CCol,Crow);
     Window(2,5,79,20);
     ClrScr;
     CheckWorkFile;
     If NewFile = ' ' Then Exit;
     CheckResultFile;
     If ((Ext3 = '.TSR') OR (EXT3 = '.tsr')) Then Exit;
     {$I-} ChDir(WorkFileDir) {$I+};
     CheckForError;
     Reset(WorkFile);
     Repeat
       ClrScr;
       CursorOn;
       Repeat
         Write('Rekordnommer waar rekords bygevoeg wil word?
-> ');
         {$I-} ReadLn(x) {$I+};
         ClearError;
         ClrScr;
         Checkdata;
       Until ErrorCode = 0;
       CursorOff;
       If x >= (FileSize(WorkFile)+1) Then
         Begin
              WriteError(' Rekord nommer val buite bestek van
lêer.');
              SoundError;
         End;
     Until x < (FileSize(WorkFile)+1);
     s := FileSize(WorkFile);
     {$I-} ChDir(WorkFileDir) {$I+};
     CheckForError;
     Close(WorkFile);
     ChDir(ProgramDir);
     AddRecords;
     {$I-} ChDir(WorkFileDir) {$I+};
     CheckForError;
     Assign(TempFile,'Temp.TSD');
     Rewrite(TempFile);
     Reset(WorkFile);
     With TestRec Do
       Begin
            For r := 0 To (x-1) Do
              Begin
                   Read(WorkFile,TestRec);
                   Write(TempFile,TestRec);
              End;
            Seek(TempFile,FileSize(TempFile));
            Seek(WorkFile,s);
            For r := s To (FileSize(WorkFile)-1) Do
              Begin
                   Read(WorkFile,TestRec);
                   Write(TempFile,TestRec);
              End;
            Seek(TempFile,FileSize(TempFile));
            Seek(WorkFile,x);
```

```
                For r := x To (s-1) Do
                  Begin
                       Read(WorkFile,TestRec);
                       Write(TempFile,TestRec);
                  End;
          End;
        Close(WorkFile);
        Erase(Workfile);
        Close(TempFile);
        Rename(TempFile,NewFile);
        ChDir(ProgramDir);
        RestoreScreen(CCol,CRow);
End;

End. {unit}
```

BYLAAG D

ATSMENU.MNU (stoor menu opsie inligting)

```
001*H*TOETS UITVOER
001*H*SKEP NUWE LêER
001*H*VERANDER BESTAANDE LêER
002Kies lêer
002Maak drukstuk
002Gebruikersopsies
002Voer toets uit
003Kies Lêer
003Identifiseer I/U Poorte
003Voeg Rekords by
003Vertoon Rekords
004Kies Lêer
004Verander Rekord
004Verwyder Rekord/s
004Voeg Rekord/s by
```

Toetsstel vir KSI kring

0	ADDR	VEKT:	131	RES:	128	SLAAG:		FAAL:		
1	CTRL	VEKT:	436	RES:	2	SLAAG:		FAAL:		
2	O_PRT	VEKT:	432	RES:	0	SLAAG:		FAAL:		
3	I_PRT	VEKT:	433	RES:	7	SLAAG:		FAAL:		
4	1t3	VEKT:	3	RES:	6	SLAAG:	1t7	FAAL:	2t	
5	1t7	VEKT:	7	RES:	0	SLAAG:	1t10	FAAL:	3t1	
6	1t10	VEKT:	10	RES:	2	SLAAG:	1t12	FAAL:	4t1	
7	1t12	VEKT:	12	RES:	1	SLAAG:	1t15	FAAL:	5t1	
8	1t15	VEKT:	15	RES:	7	SLAAG:	f0	FAAL:		
9	2t7	VEKT:	7	RES:	0	SLAAG:	2t10	FAAL:	6t1	
10	2t10	VEKT:	10	RES:	2	SLAAG:	2t12	FAAL:	7t1	
11	2t12	VEKT:	12	RES:	1	SLAAG:	f19	FAAL:	f1	
12	3t10	VEKT:	10	RES:	2	SLAAG:	3t12	FAAL:	8t1	
13	3t12	VEKT:	12	RES:	1	SLAAG:	f14	FAAL:	9t1	
14	4t12	VEKT:	12	RES:	1	SLAAG:	f9	FAAL:	10t1	
15	5t15	VEKT:	15	RES:	7	SLAAG:	f4	FAAL:	f1	
16	6t10	VEKT:	10	RES:	2	SLAAG:	6t12	FAAL:	11t1	
17	6t12	VEKT:	12	RES:	1	SLAAG:	6t15	FAAL:	f1	
18	6t15	VEKT:	15	RES:	7	SLAAG:	f8	FAAL:	f	
19	7t12	VEKT:	12	RES:	1	SLAAG:	7t15	FAAL:	f1	
20	7t15	VEKT:	15	RES:	7	SLAAG:	f6	FAAL:	f1	
21	8t12	VEKT:	12	RES:	1	SLAAG:	f12	FAAL:	f2	
22	9t15	VEKT:	15	RES:	7	SLAAG:	f18	FAAL:	f	
23	10t15	VEKT:	15	RES:	7	SLAAG:	f2	FAAL:	f	
24	11t15	VEKT:	15	RES:	7	SLAAG:	f16	FAAL:	f	
25	END	VEKT:	0	RES:	0	SLAAG:		FAAL:		
26	f0	VEKT:	1	RES:	0	SLAAG:		FAAL:		
27	FFREE	VEKT:	0	RES:	0	SLAAG:		FAAL:		
28	f1	VEKT:	1	RES:	0	SLAAG:		FAAL:		
29	DATA0/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
30	u3-13/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
31	f2	VEKT:	1	RES:	0	SLAAG:		FAAL:		
32	DATA0/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		
33	u3-13/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		
34	f3	VEKT:	1	RES:	0	SLAAG:		FAAL:		
35	DATA1/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
36	u3-10/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
37	f4	VEKT:	1	RES:	0	SLAAG:		FAAL:		
38	DATA1/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		
39	u3-10/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		
40	f5	VEKT:	1	RES:	0	SLAAG:		FAAL:		
41	DATA2/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
42	u3-5/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
43	f6	VEKT:	1	RES:	0	SLAAG:		FAAL:		
44	DATA2/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		
45	u3-5/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		
46	f7	VEKT:	1	RES:	0	SLAAG:		FAAL:		
47	DATA3/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
48	u2-5/0	VEKT:	0	RES:	0	SLAAG:	u3-2/0	FAAL:		
49	f8	VEKT:	1	RES:	0	SLAAG:		FAAL:		
50	DATA3/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		
51	u2-5/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		
52	f9	VEKT:	1	RES:	0	SLAAG:		FAAL:		
53	u1-3/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
54	f10	VEKT:	1	RES:	0	SLAAG:		FAAL:		
55	u1-3/1	VEKT:	0	RES:	0	SLAAG:		FAAL:		

56	f11	VEKT:		0	RES:	0	SLAAG:			FAAL:	
57	u1-8/0	VEKT:		0	RES:	0	SLAAG:			FAAL:	
58	f12	VEKT:		1	RES:	0	SLAAG:			FAAL:	
59	u1-8/1	VEKT:		0	RES:	0	SLAAG:			FAAL:	
60	f13	VEKT:		1	RES:	0	SLAAG:			FAAL:	
61	u2-3/0	VEKT:		0	RES:	0	SLAAG:			FAAL:	
62	f14	VEKT:		1	RES:	0	SLAAG:			FAAL:	
63	u2-3/1	VEKT:		0	RES:	0	SLAAG:			FAAL:	
64	f15	VEKT:		1	RES:	0	SLAAG:			FAAL:	
65	u1-6/0	VEKT:		0	RES:	0	SLAAG:			FAAL:	
66	f16	VEKT:		1	RES:	0	SLAAG:			FAAL:	
67	u1-6/1	VEKT:		0	RES:	0	SLAAG:			FAAL:	
68	f17	VEKT:		1	RES:	0	SLAAG:			FAAL:	
69	u1-11/0	VEKT:		0	RES:	0	SLAAG:			FAAL:	
70	f18	VEKT:		1	RES:	0	SLAAG:			FAAL:	
71	u1-11/1	VEKT:		0	RES:	0	SLAAG:			FAAL:	
72	f19	VEKT:		1	RES:	0	SLAAG:			FAAL:	
73	u2-6/0	VEKT:		0	RES:	0	SLAAG:			FAAL:	
74	f20	VEKT:		1	RES:	0	SLAAG:			FAAL:	
75	u2-6/1	VEKT:		0	RES:	0	SLAAG:			FAAL:	
76	END	VEKT:		0	RES:	0	SLAAG:			FAAL:	
77	CTRL	VEKT:		436	RES:	11	SLAAG:			FAAL:	
78	O_PRT	VEKT:		432	RES:	0	SLAAG:			FAAL:	
79	I_PRT	VEKT:		433	RES:	56	SLAAG:			FAAL:	
80	1t1	VEKT:		1	RES:	24	SLAAG:	1t2	FAAL:		2t
81	1t2	VEKT:		2	RES:	40	SLAAG:	1t4	FAAL:		3t
82	1t4	VEKT:		4	RES:	48	SLAAG:	1t8	FAAL:		4t
83	1t8	VEKT:		8	RES:	56	SLAAG:	1t16	FAAL:		f
84	1t16	VEKT:		16	RES:	8	SLAAG:	1t32	FAAL:		f1
85	1t32	VEKT:		32	RES:	16	SLAAG:	1t64	FAAL:		f1
86	1t64	VEKT:		64	RES:	32	SLAAG:	f0	FAAL:		f
87	2t2	VEKT:		2	RES:	40	SLAAG:	2t4	FAAL:		5t
88	2t4	VEKT:		4	RES:	48	SLAAG:	2t8	FAAL:		6t
89	2t8	VEKT:		8	RES:	56	SLAAG:	2t16	FAAL:		f1
90	2t16	VEKT:		16	RES:	8	SLAAG:	f7	FAAL:		7t6
91	3t4	VEKT:		4	RES:	48	SLAAG:	3t8	FAAL:		8t
92	3t8	VEKT:		8	RES:	56	SLAAG:	3t16	FAAL:		f1
93	3t16	VEKT:		16	RES:	8	SLAAG:	f5	FAAL:		9t3
94	4t8	VEKT:		8	RES:	56	SLAAG:	4t16	FAAL:		f2
95	4t16	VEKT:		16	RES:	8	SLAAG:	4t32	FAAL:		f2
96	4t32	VEKT:		32	RES:	16	SLAAG:	f3	FAAL:		f3
97	5t4	VEKT:		4	RES:	48	SLAAG:	5t8	FAAL:		10t
98	5t8	VEKT:		8	RES:	56	SLAAG:	f19	FAAL:		11t1
99	6t8	VEKT:		8	RES:	56	SLAAG:	f17	FAAL:		12t1
100	7t64	VEKT:		64	RES:	32	SLAAG:	f28	FAAL:		f2
101	8t8	VEKT:		8	RES:	56	SLAAG:	f15	FAAL:		13t1
102	9t32	VEKT:		32	RES:	16	SLAAG:	f30	FAAL:		f2
103	10t8	VEKT:		8	RES:	56	SLAAG:	f2	FAAL:		14t1
104	11t16	VEKT:		16	RES:	8	SLAAG:	f25	FAAL:		15t3
105	12t16	VEKT:		16	RES:	8	SLAAG:	12t32	FAAL:		f
106	12t32	VEKT:		32	RES:	16	SLAAG:	f23	FAAL:		f2
107	13t16	VEKT:		16	RES:	8	SLAAG:	13t64	FAAL:		f
108	13t64	VEKT:		64	RES:	32	SLAAG:	f21	FAAL:		f2
109	14t16	VEKT:		16	RES:	8	SLAAG:	f14	FAAL:		16t3
110	15t32	VEKT:		32	RES:	16	SLAAG:	f31	FAAL:		f
111	16t32	VEKT:		32	RES:	16	SLAAG:	f12	FAAL:		f1

```
112      END VEKT:     1 RES:     0 SLAAG:          FAAL:
113       f0 VEKT:     1 RES:     0 SLAAG:          FAAL:
114    FFREE VEKT:     0 RES:     0 SLAAG:          FAAL:
115       f1 VEKT:     1 RES:     0 SLAAG:          FAAL:
116 u5-13/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
117       f2 VEKT:     1 RES:     0 SLAAG:          FAAL:
118 u5-13/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
119       f3 VEKT:     1 RES:     0 SLAAG:          FAAL:
120  u5-2/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
121       f4 VEKT:     1 RES:     0 SLAAG:          FAAL:
122  u5-2/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
123       f5 VEKT:     1 RES:     0 SLAAG:          FAAL:
124  u5-5/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
125       f6 VEKT:     1 RES:     0 SLAAG:          FAAL:
126  u5-5/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
127       f7 VEKT:     1 RES:     0 SLAAG:          FAAL:
128 u5-10/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
129       f8 VEKT:     1 RES:     0 SLAAG:          FAAL:
130 u5-10/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
131       f9 VEKT:     1 RES:     0 SLAAG:          FAAL:
132  u7-2/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
133      f10 VEKT:     1 RES:     0 SLAAG:          FAAL:
134  u7-2/1 VEKT:     0 RES:     0 SLAAG:   U4-1/0 FAAL:
135      f11 VEKT:     1 RES:     0 SLAAG:          FAAL:
136 U7-13/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
137      f12 VEKT:     1 RES:     0 SLAAG:          FAAL:
138 U7-13/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
139      f13 VEKT:     1 RES:     0 SLAAG:          FAAL:
140 u8-10/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
141      f14 VEKT:     1 RES:     0 SLAAG:          FAAL:
142 u8-10/1 VEKT:     0 RES:     0 SLAAG:  u3-12/0 FAAL:
143      f15 VEKT:     1 RES:     0 SLAAG:          FAAL:
144  u2-8/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
145      f16 VEKT:     1 RES:     0 SLAAG:          FAAL:
146  u2-8/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
147      f17 VEKT:     1 RES:     0 SLAAG:          FAAL:
148  u7-6/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
149      f18 VEKT:     1 RES:     0 SLAAG:          FAAL:
150  u7-6/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
151      f19 VEKT:     1 RES:     0 SLAAG:          FAAL:
152  u8-3/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
153      f20 VEKT:     1 RES:     0 SLAAG:          FAAL:
154  u8-3/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
155      f21 VEKT:     1 RES:     0 SLAAG:          FAAL:
156 u2-11/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
157      f22 VEKT:     1 RES:     0 SLAAG:          FAAL:
158 u2-11/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
159      f23 VEKT:     1 RES:     0 SLAAG:          FAAL:
160  u7-8/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
161      f24 VEKT:     1 RES:     0 SLAAG:          FAAL:
162  u7-8/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
163      f25 VEKT:     1 RES:     0 SLAAG:          FAAL:
164  u8-6/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
165      f26 VEKT:     1 RES:     0 SLAAG:          FAAL:
166  u8-6/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
167      f27 VEKT:     1 RES:     0 SLAAG:          FAAL:
```

```
168   u7-3/0 VEKT:          0 RES:      0 SLAAG:            FAAL:
169      f28 VEKT:          1 RES:      0 SLAAG:            FAAL:
170   u7-3/1 VEKT:          0 RES:      0 SLAAG:            FAAL:
171      f29 VEKT:          1 RES:      0 SLAAG:            FAAL:
172  u7-11/0 VEKT:          0 RES:      0 SLAAG:            FAAL:
173      f30 VEKT:          1 RES:      0 SLAAG:            FAAL:
174  u7-11/1 VEKT:          0 RES:      0 SLAAG:            FAAL:
175      f31 VEKT:          1 RES:      0 SLAAG:            FAAL:
176   u8-8/0 VEKT:          0 RES:      0 SLAAG:            FAAL:
177      f32 VEKT:          1 RES:      0 SLAAG:            FAAL:
178   u8-8/1 VEKT:          0 RES:      0 SLAAG:            FAAL:
179      END VEKT:          0 RES:      0 SLAAG:            FAAL:
180     CTRL VEKT:        436 RES:     15 SLAAG:            FAAL:
181    O_PRT VEKT:        432 RES:      0 SLAAG:            FAAL:
182    O_PRT VEKT:        434 RES:      0 SLAAG:            FAAL:
183    I_PRT VEKT:        434 RES:     15 SLAAG:            FAAL:
184      1t2 VEKT:          2 RES:      0 SLAAG:            FAAL:
185      1t2 VEKT:          0 RES:      2 SLAAG:     1t17   FAAL:     2t1
186     1t17 VEKT:          1 RES:      0 SLAAG:            FAAL:
187     1t17 VEKT:         16 RES:      1 SLAAG:     1t48   FAAL:     3t4
188     1t48 VEKT:          0 RES:      0 SLAAG:            FAAL:
189     1t48 VEKT:         48 RES:      1 SLAAG:     1t49   FAAL:     4t4
190     1t49 VEKT:          1 RES:      0 SLAAG:            FAAL:
191     1t49 VEKT:         48 RES:      0 SLAAG:     1t68   FAAL:      f2
192     1t68 VEKT:          4 RES:      0 SLAAG:            FAAL:
193     1t68 VEKT:         64 RES:      4 SLAAG:     1t80   FAAL:     5t9
194     1t80 VEKT:          0 RES:      0 SLAAG:            FAAL:
195     1t80 VEKT:         80 RES:      2 SLAAG:     1t82   FAAL:     6t8
196     1t82 VEKT:          2 RES:      0 SLAAG:            FAAL:
197     1t82 VEKT:         80 RES:      0 SLAAG:     1t96   FAAL:      f5
198     1t96 VEKT:          0 RES:      0 SLAAG:            FAAL:
199     1t96 VEKT:         96 RES:      4 SLAAG:    1t100   FAAL:    7t10
200    1t100 VEKT:          4 RES:      0 SLAAG:            FAAL:
201    1t100 VEKT:         96 RES:      0 SLAAG:    1t104   FAAL:      f3
202    1t104 VEKT:          8 RES:      0 SLAAG:            FAAL:
203    1t104 VEKT:         96 RES:     12 SLAAG:    1t112   FAAL:    8t11
204    1t112 VEKT:          0 RES:      0 SLAAG:            FAAL:
205    1t112 VEKT:        112 RES:      8 SLAAG:    1t113   FAAL:    9t12
206    1t113 VEKT:          1 RES:      0 SLAAG:            FAAL:
207    1t113 VEKT:        113 RES:      9 SLAAG:    1t114   FAAL:
208    1t114 VEKT:          2 RES:      0 SLAAG:            FAAL:
209    1t114 VEKT:        113 RES:     10 SLAAG:    1t116   FAAL:
210    1t116 VEKT:          4 RES:      0 SLAAG:            FAAL:
211    1t116 VEKT:        112 RES:     12 SLAAG:    1t120   FAAL:
212    1t120 VEKT:          8 RES:      0 SLAAG:            FAAL:
213    1t120 VEKT:        112 RES:      0 SLAAG:       f0   FAAL:      f4
214     2t17 VEKT:          1 RES:      0 SLAAG:            FAAL:
215     2t17 VEKT:         16 RES:      1 SLAAG:     2t48   FAAL:    10t4
216     2t48 VEKT:          0 RES:      0 SLAAG:            FAAL:
217     2t48 VEKT:         48 RES:      1 SLAAG:     2t49   FAAL:      f2
218     2t49 VEKT:          1 RES:      0 SLAAG:            FAAL:
219     2t49 VEKT:         48 RES:      0 SLAAG:     2t80   FAAL:      f3
220     2t80 VEKT:          0 RES:      0 SLAAG:            FAAL:
221     2t80 VEKT:         80 RES:      2 SLAAG:     2t82   FAAL:      f3
222     2t82 VEKT:          2 RES:      0 SLAAG:            FAAL:
223     2t82 VEKT:         80 RES:      0 SLAAG:      f39   FAAL:      f2
```

224	3t48	VEKT:	0	RES:	0	SLAAG:		FAAL:		
225	3t48	VEKT:	48	RES:	1	SLAAG:	3t49	FAAL:	11t4	
226	3t49	VEKT:	1	RES:	0	SLAAG:		FAAL:		
227	3t49	VEKT:	48	RES:	0	SLAAG:	3t68	FAAL:	f1	
228	3t68	VEKT:	4	RES:	0	SLAAG:		FAAL:		
229	3t68	VEKT:	64	RES:	4	SLAAG:	f27	FAAL:	f	
230	4t49	VEKT:	1	RES:	0	SLAAG:		FAAL:		
231	4t49	VEKT:	48	RES:	0	SLAAG:	f29	FAAL:	12t8	
232	5t96	VEKT:	0	RES:	0	SLAAG:		FAAL:		
233	5t96	VEKT:	96	RES:	4	SLAAG:	5t100	FAAL:	13t10	
234	5t100	VEKT:	4	RES:	0	SLAAG:		FAAL:		
235	5t100	VEKT:	96	RES:	0	SLAAG:	f36	FAAL:	f2	
236	6t82	VEKT:	2	RES:	0	SLAAG:		FAAL:		
237	6t82	VEKT:	80	RES:	0	SLAAG:	f40	FAAL:	14t9	
238	7t100	VEKT:	4	RES:	0	SLAAG:		FAAL:		
239	7t100	VEKT:	96	RES:	0	SLAAG:	f35	FAAL:	f	
240	8t112	VEKT:	0	RES:	0	SLAAG:		FAAL:		
241	8t112	VEKT:	112	RES:	8	SLAAG:	8t120	FAAL:	f4	
242	8t120	VEKT:	8	RES:	0	SLAAG:		FAAL:		
243	8t120	VEKT:	112	RES:	0	SLAAG:	f43	FAAL:	f2	
244	9t120	VEKT:	8	RES:	0	SLAAG:		FAAL:		
245	9t120	VEKT:	112	RES:	0	SLAAG:	f41	FAAL:	f	
246	10t48	VEKT:	0	RES:	0	SLAAG:		FAAL:		
247	10t48	VEKT:	48	RES:	1	SLAAG:	f14	FAAL:	15t6	
248	11t49	VEKT:	1	RES:	0	SLAAG:		FAAL:		
249	11t49	VEKT:	48	RES:	0	SLAAG:	f46	FAAL:	16t6	
250	12t80	VEKT:	0	RES:	0	SLAAG:		FAAL:		
251	12t80	VEKT:	80	RES:	2	SLAAG:	12t96	FAAL:	f1	
252	12t96	VEKT:	0	RES:	0	SLAAG:		FAAL:		
253	12t96	VEKT:	96	RES:	4	SLAAG:	f11	FAAL:	f3	
254	13t100	VEKT:	4	RES:	0	SLAAG:		FAAL:		
255	13t100	VEKT:	96	RES:	0	SLAAG:	f32	FAAL:	f2	
256	14t96	VEKT:	0	RES:	0	SLAAG:		FAAL:		
257	14t96	VEKT:	96	RES:	4	SLAAG:	f9	FAAL:	f5	
258	15t68	VEKT:	4	RES:	0	SLAAG:		FAAL:		
259	15t68	VEKT:	64	RES:	4	SLAAG:	15t96	FAAL:	17t8	
260	15t96	VEKT:	0	RES:	0	SLAAG:		FAAL:		
261	15t96	VEKT:	96	RES:	4	SLAAG:	f34	FAAL:	f24	
262	16t68	VEKT:	4	RES:	0	SLAAG:		FAAL:		
263	16t68	VEKT:	64	RES:	4	SLAAG:	f6	FAAL:	18t8	
264	17t80	VEKT:	0	RES:	0	SLAAG:		FAAL:		
265	17t80	VEKT:	80	RES:	2	SLAAG:	f16	FAAL:	19t96	
266	18t80	VEKT:	0	RES:	0	SLAAG:		FAAL:		
267	18t80	VEKT:	80	RES:	2	SLAAG:	f38	FAAL:	f22	
268	19t96	VEKT:	0	RES:	0	SLAAG:		FAAL:		
269	19t96	VEKT:	96	RES:	4	SLAAG:	f50	FAAL:	20t104	
270	20t104	VEKT:	8	RES:	0	SLAAG:		FAAL:		
271	20t104	VEKT:	96	RES:	12	SLAAG:	20t112	FAAL:	f18	
272	20t112	VEKT:	0	RES:	0	SLAAG:		FAAL:		
273	20t112	VEKT:	112	RES:	8	SLAAG:	f45	FAAL:	f26	
274	END	VEKT:	0	RES:	0	SLAAG:		FAAL:		
275	f0	VEKT:	1	RES:	0	SLAAG:		FAAL:		
276	FFREE	VEKT:	0	RES:	0	SLAAG:		FAAL:		
277	f1	VEKT:	1	RES:	0	SLAAG:		FAAL:		
278	u10-5/0	VEKT:	0	RES:	0	SLAAG:		FAAL:		
279	f2	VEKT:	1	RES:	0	SLAAG:		FAAL:		

```
280 u10-5/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
281      f3 VEKT:     1 RES:     0 SLAAG:          FAAL:
282 u10-3/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
283      f4 VEKT:     1 RES:     0 SLAAG:          FAAL:
284 u10-3/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
285      f5 VEKT:     1 RES:     0 SLAAG:          FAAL:
286 u10-1/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
287      f6 VEKT:     1 RES:     0 SLAAG:          FAAL:
288 u10-1/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
289      f7 VEKT:     1 RES:     0 SLAAG:          FAAL:
290 u10-6/0 VEKT:     0 RES:     0 SLAAG:     y6/0 FAAL:
291      f8 VEKT:     2 RES:     0 SLAAG:          FAAL:
292 u10-6/1 VEKT:     0 RES:     0 SLAAG:  u10-4/1 FAAL:  u10-2/
293    y7/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
294      f9 VEKT:     1 RES:     0 SLAAG:          FAAL:
295 u10-4/0 VEKT:     0 RES:     0 SLAAG:     y5/0 FAAL:
296     f11 VEKT:     1 RES:     0 SLAAG:          FAAL:
297 u10-2/0 VEKT:     0 RES:     0 SLAAG:     y3/0 FAAL:
298     f14 VEKT:     1 RES:     0 SLAAG:          FAAL:
299    y3/1 VEKT:     0 RES:     0 SLAAG:  u5-12/0 FAAL:
300     f16 VEKT:     1 RES:     0 SLAAG:          FAAL:
301    y5/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
302     f18 VEKT:     1 RES:     0 SLAAG:          FAAL:
303    y7/1 VEKT:     0 RES:     0 SLAAG:  u6-13/0 FAAL:
304     f19 VEKT:     1 RES:     0 SLAAG:          FAAL:
305 u10-9/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
306     f20 VEKT:     1 RES:     0 SLAAG:          FAAL:
307 u10-9/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
308     f21 VEKT:     1 RES:     0 SLAAG:          FAAL:
309 u15-1/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
310     f22 VEKT:     1 RES:     0 SLAAG:          FAAL:
311 u15-1/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
312     f23 VEKT:     1 RES:     0 SLAAG:          FAAL:
313 u14-2/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
314     f24 VEKT:     1 RES:     0 SLAAG:          FAAL:
315 u14-2/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
316     f25 VEKT:     1 RES:     0 SLAAG:          FAAL:
317 u11-3/0 VEKT:     0 RES:     0 SLAAG:          FAAL:
318     f26 VEKT:     1 RES:     0 SLAAG:          FAAL:
319 u11-3/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
320     f27 VEKT:     1 RES:     0 SLAAG:          FAAL:
321u10-10/0 VEKT:     0 RES:     0 SLAAG:   u9-8/0 FAAL:
322     f28 VEKT:     1 RES:     0 SLAAG:          FAAL:
323u10-10/1 VEKT:     0 RES:     0 SLAAG:  u10-8/1 FAAL:
324     f29 VEKT:     1 RES:     0 SLAAG:          FAAL:
325 u10-8/0 VEKT:     0 RES:     0 SLAAG:  u9-11/0 FAAL:
326     f30 VEKT:     1 RES:     0 SLAAG:          FAAL:
327 u10-8/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
328     f31 VEKT:     1 RES:     0 SLAAG:          FAAL:
329 u9-11/1 VEKT:     0 RES:     0 SLAAG:   u9-8/1 FAAL:  u16-3/
330     f32 VEKT:     1 RES:     0 SLAAG:          FAAL:
331 u14-3/0 VEKT:     0 RES:     0 SLAAG: u14-11/0 FAAL:
332     f33 VEKT:     1 RES:     0 SLAAG:          FAAL:
333 u14-3/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
334     f34 VEKT:     1 RES:     0 SLAAG:          FAAL:
335 u14-6/0 VEKT:     0 RES:     0 SLAAG:  u14-8/0 FAAL: u14-11/
```

```
336      f35 VEKT:      0 RES:      0 SLAAG:            FAAL:
337  u14-6/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
338      f36 VEKT:      1 RES:      0 SLAAG:            FAAL:
339  u14-8/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
340      f37 VEKT:      1 RES:      0 SLAAG:            FAAL:
341  u15-3/0 VEKT:      0 RES:      0 SLAAG:  u15-11/0  FAAL:
342      f38 VEKT:      1 RES:      0 SLAAG:            FAAL:
343  u15-6/0 VEKT:      0 RES:      0 SLAAG:   u15-8/0  FAAL:  u15-11/1
344      f39 VEKT:      1 RES:      0 SLAAG:            FAAL:
345  u15-6/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
346      f40 VEKT:      1 RES:      0 SLAAG:            FAAL:
347  u15-8/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
348      f41 VEKT:      1 RES:      0 SLAAG:            FAAL:
349  u11-4/0 VEKT:      0 RES:      0 SLAAG:    u9-3/0  FAAL:
350      f42 VEKT:      1 RES:      0 SLAAG:            FAAL:
351  u11-4/1 VEKT:      0 RES:      0 SLAAG:   u11-2/1  FAAL:
352      f43 VEKT:      1 RES:      0 SLAAG:            FAAL:
353  u11-2/0 VEKT:      0 RES:      0 SLAAG:    u9-6/0  FAAL:
354      f44 VEKT:      1 RES:      0 SLAAG:            FAAL:
355  u11-2/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
356      f45 VEKT:      1 RES:      0 SLAAG:            FAAL:
357   u9-3/1 VEKT:      0 RES:      0 SLAAG:    u9-6/1  FAAL:  u16-6/1
358      f46 VEKT:      1 RES:      0 SLAAG:            FAAL:
359  u16-3/0 VEKT:      0 RES:      0 SLAAG:            FAAL:
360      f47 VEKT:      1 RES:      0 SLAAG:            FAAL:
361u15-11/0 VEKT:      0 RES:      0 SLAAG:            FAAL:
362      f48 VEKT:      1 RES:      0 SLAAG:            FAAL:
363u14-11/0 VEKT:      0 RES:      0 SLAAG:            FAAL:
364      f49 VEKT:      1 RES:      0 SLAAG:            FAAL:
365  u16-6/0 VEKT:      0 RES:      0 SLAAG:            FAAL:
366      f50 VEKT:      1 RES:      0 SLAAG:            FAAL:
367     y6/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
368      f51 VEKT:      1 RES:      0 SLAAG:            FAAL:
369  u15-3/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
370      END VEKT:      0 RES:      0 SLAAG:            FAAL:
371     CTRL VEKT:    436 RES:      9 SLAAG:            FAAL:
372    O_PRT VEKT:    432 RES:      0 SLAAG:            FAAL:
373    I_PRT VEKT:    433 RES:     56 SLAAG:            FAAL:
374       t1 VEKT:     15 RES:      0 SLAAG:        f0  FAAL:        f1
375      END VEKT:      0 RES:      0 SLAAG:            FAAL:
376       f0 VEKT:      1 RES:      0 SLAAG:            FAAL:
377    FFREE VEKT:      0 RES:      0 SLAAG:            FAAL:
378       f1 VEKT:      1 RES:      0 SLAAG:            FAAL:
379   u4-1/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
380      END VEKT:      0 RES:      0 SLAAG:            FAAL:
381     CTRL VEKT:    436 RES:     11 SLAAG:            FAAL:
382    O_PRT VEKT:    432 RES:      0 SLAAG:            FAAL:
383    O_PRT VEKT:    434 RES:      0 SLAAG:            FAAL:
384    I_PRT VEKT:    434 RES:     15 SLAAG:            FAAL:
385       t1 VEKT:     15 RES:      0 SLAAG:            FAAL:
386       t1 VEKT:      0 RES:      0 SLAAG:        f0  FAAL:        f:
387      END VEKT:      0 RES:      0 SLAAG:            FAAL:
388       f0 VEKT:      1 RES:      0 SLAAG:            FAAL:
389    FFREE VEKT:      0 RES:      0 SLAAG:            FAAL:
390       f1 VEKT:      1 RES:      0 SLAAG:            FAAL:
391  u5-12/1 VEKT:      0 RES:      0 SLAAG:            FAAL:
```

```
392      END VEKT:     0 RES:     0 SLAAG:           FAAL:
393     CTRL VEKT:   436 RES:     6 SLAAG:           FAAL:
394   O_PRT VEKT:   432 RES:     0 SLAAG:           FAAL:
395   O_PRT VEKT:   434 RES:     0 SLAAG:           FAAL:
396   I_PRT VEKT:   434 RES:    15 SLAAG:           FAAL:
397       t1 VEKT:    15 RES:     0 SLAAG:           FAAL:
398       t1 VEKT:   112 RES:     0 SLAAG:        f0 FAAL:        f:
399      END VEKT:     0 RES:     0 SLAAG:           FAAL:
400       f0 VEKT:     1 RES:     0 SLAAG:           FAAL:
401    FFREE VEKT:     0 RES:     0 SLAAG:           FAAL:
402       f1 VEKT:     1 RES:     0 SLAAG:           FAAL:
403 u6-13/1 VEKT:     0 RES:     0 SLAAG:  u3-12/1 FAAL:
404      END VEKT:     0 RES:     0 SLAAG:           FAAL:
405      END VEKT:     0 RES:     0 SLAAG:           FAAL:
```

Toetsstel vir MSI kring

0	ADDR	VEKT:	130	RES:	130	SLAAG:		FAAL:	
1	O_PRT	VEKT:	432	RES:	0	SLAAG:		FAAL:	
2	O_PRT	VEKT:	434	RES:	0	SLAAG:		FAAL:	
3	I_PRT	VEKT:	433	RES:	64	SLAAG:		FAAL:	
4	t1	VEKT:	6	RES:	0	SLAAG:		FAAL:	
5	t1	VEKT:	0	RES:	0	SLAAG:	t2	FAAL:	f
6	t2	VEKT:	6	RES:	0	SLAAG:		FAAL:	
7	t2	VEKT:	32	RES:	64	SLAAG:	f0	FAAL:	f
8	END	VEKT:	0	RES:	0	SLAAG:		FAAL:	
9	f0	VEKT:	1	RES:	0	SLAAG:		FAAL:	
10	FFREE	VEKT:	0	RES:	0	SLAAG:		FAAL:	
11	f1	VEKT:	1	RES:	0	SLAAG:		FAAL:	
12	u10-1/1	VEKT:	0	RES:	0	SLAAG:	u14-3/1	FAAL:	
13	f2	VEKT:	1	RES:	0	SLAAG:		FAAL:	
14	u10-1/0	VEKT:	0	RES:	0	SLAAG:	u14-3/0	FAAL:	u14-4/
15	END	VEKT:	0	RES:	0	SLAAG:		FAAL:	
16	CTRL	VEKT:	434	RES:	32	SLAAG:		FAAL:	
17	O_PRT	VEKT:	432	RES:	0	SLAAG:		FAAL:	
18	I_PRT	VEKT:	433	RES:	32	SLAAG:		FAAL:	
19	I_PRT	VEKT:	433	RES:	16	SLAAG:		FAAL:	
20	t1	VEKT:	6	RES:	0	SLAAG:	t2	FAAL:	f
21	t1	VEKT:	0	RES:	0	SLAAG:	t2	FAAL:	f
22	t2	VEKT:	14	RES:	0	SLAAG:	t3	FAAL:	f
23	t2	VEKT:	0	RES:	0	SLAAG:	t3	FAAL:	
24	t3	VEKT:	10	RES:	32	SLAAG:	t4	FAAL:	f
25	t3	VEKT:	0	RES:	16	SLAAG:	t4	FAAL:	f
26	t4	VEKT:	14	RES:	32	SLAAG:	CLK	FAAL:	
27	t4	VEKT:	0	RES:	16	SLAAG:	CLK	FAAL:	
28	CLK	VEKT:	434	RES:	0	SLAAG:		FAAL:	
29	CLK	VEKT:	1	RES:	32	SLAAG:		FAAL:	
30	t6	VEKT:	15	RES:	32	SLAAG:	t7	FAAL:	f
31	t6	VEKT:	0	RES:	16	SLAAG:	t7	FAAL:	
32	t7	VEKT:	7	RES:	0	SLAAG:	f0	FAAL:	f
33	t7	VEKT:	0	RES:	0	SLAAG:	f0	FAAL:	
34	END	VEKT:	0	RES:	0	SLAAG:		FAAL:	
35	f0	VEKT:	1	RES:	0	SLAAG:		FAAL:	
36	FFREE	VEKT:	0	RES:	0	SLAAG:		FAAL:	
37	f1	VEKT:	1	RES:	0	SLAAG:		FAAL:	
38	u11-3/1	VEKT:	0	RES:	0	SLAAG:		FAAL:	
39	f2	VEKT:	1	RES:	0	SLAAG:		FAAL:	
40	u11-6/1	VEKT:	0	RES:	0	SLAAG:		FAAL:	
41	f3	VEKT:	1	RES:	0	SLAAG:		FAAL:	
42	u10-5/1	VEKT:	0	RES:	0	SLAAG:	u10-4/0	FAAL:	
43	f4	VEKT:	1	RES:	0	SLAAG:		FAAL:	
44	u11-1/1	VEKT:	0	RES:	0	SLAAG:		FAAL:	
45	f5	VEKT:	2	RES:	0	SLAAG:		FAAL:	
46	u11-3/0	VEKT:	0	RES:	0	SLAAG:	u11-1/0	FAAL:	u10-5/
47	u10-4/1	VEKT:	0	RES:	0	SLAAG:		FAAL:	
48	f6	VEKT:	1	RES:	0	SLAAG:		FAAL:	
49	u11-6/0	VEKT:	0	RES:	0	SLAAG:	u10-9/0	FAAL:	u10-12/
50	f7	VEKT:	1	RES:	0	SLAAG:		FAAL:	
51	u10-3/0	VEKT:	0	RES:	0	SLAAG:	u10-3/1	FAAL:	
52	END	VEKT:	0	RES:	0	SLAAG:		FAAL:	
53	O_PRT	VEKT:	434	RES:	0	SLAAG:		FAAL:	
54	I_PRT	VEKT:	433	RES:	128	SLAAG:		FAAL:	
55	CLK	VEKT:	434	RES:	0	SLAAG:		FAAL:	

```
 56      CLK VEKT:     1 RES:    32 SLAAG:           FAAL:
 57      CLK VEKT:   432 RES:     1 SLAAG:           FAAL:
 58      CLK VEKT:     1 RES:     0 SLAAG:           FAAL:
 59       t1 VEKT:     0 RES:     0 SLAAG:        t2 FAAL:        f
 60       t2 VEKT:    48 RES:     0 SLAAG:        t3 FAAL:        f
 61       t3 VEKT:    32 RES:     0 SLAAG:       CLK FAAL:       CL
 62      CLK VEKT:   432 RES:     7 SLAAG:           FAAL:
 63      CLK VEKT:     1 RES:     6 SLAAG:           FAAL:
 64      CLK VEKT:   432 RES:    14 SLAAG:           FAAL:
 65      CLK VEKT:     1 RES:     6 SLAAG:           FAAL:
 66       t4 VEKT:    32 RES:     0 SLAAG:        t5 FAAL:        f
 67       t5 VEKT:    48 RES:   128 SLAAG:        f0 FAAL:        f
 68      END VEKT:     0 RES:     0 SLAAG:           FAAL:
 69       f0 VEKT:     1 RES:     0 SLAAG:           FAAL:
 70    FFREE VEKT:     0 RES:     0 SLAAG:           FAAL:
 71       f1 VEKT:     1 RES:     0 SLAAG:           FAAL:
 72u11-11/1 VEKT:     0 RES:     0 SLAAG:           FAAL:
 73       f2 VEKT:     1 RES:     0 SLAAG:           FAAL:
 74 u10-9/1 VEKT:     0 RES:     0 SLAAG: u10-10/0 FAAL:
 75       f3 VEKT:     1 RES:     0 SLAAG:           FAAL:
 76u11-12/1 VEKT:     0 RES:     0 SLAAG:           FAAL:
 77       f4 VEKT:     1 RES:     0 SLAAG:           FAAL:
 78u11-12/0 VEKT:     0 RES:     0 SLAAG: u11-11/0 FAAL:
 79      END VEKT:     0 RES:     0 SLAAG:           FAAL:
 80    O_PRT VEKT:   434 RES:     0 SLAAG:           FAAL:
 81    I_PRT VEKT:   433 RES:     1 SLAAG:           FAAL:
 82    I_PRT VEKT:   433 RES:     2 SLAAG:           FAAL:
 83    I_PRT VEKT:   433 RES:     4 SLAAG:           FAAL:
 84    I_PRT VEKT:   433 RES:     8 SLAAG:           FAAL:
 85      CLK VEKT:   434 RES:     0 SLAAG:           FAAL:
 86      CLK VEKT:     1 RES:    32 SLAAG:           FAAL:
 87      CLK VEKT:   432 RES:   241 SLAAG:           FAAL:
 88      CLK VEKT:     1 RES:   240 SLAAG:           FAAL:
 89       t1 VEKT:    32 RES:     0 SLAAG:        t2 FAAL:        f
 90       t1 VEKT:     0 RES:     0 SLAAG:        t2 FAAL:        f
 91       t1 VEKT:     0 RES:     0 SLAAG:        t2 FAAL:        f
 92       t1 VEKT:     0 RES:     0 SLAAG:        t2 FAAL:        f
 93       t2 VEKT:    48 RES:     0 SLAAG:       CLK FAAL:
 94       t2 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
 95       t2 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
 96       t2 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
 97      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
 98      CLK VEKT:     1 RES:   246 SLAAG:           FAAL:
 99       t3 VEKT:    48 RES:     1 SLAAG:       CLK FAAL:        f
100       t3 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
101       t3 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
102       t3 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
103      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
104      CLK VEKT:     1 RES:   246 SLAAG:           FAAL:
105       t4 VEKT:    48 RES:     0 SLAAG:       CLK FAAL:
106       t4 VEKT:     0 RES:     2 SLAAG:       CLK FAAL:        f
107       t4 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
108       t4 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
109      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
110      CLK VEKT:     2 RES:   246 SLAAG:           FAAL:
111       t5 VEKT:    48 RES:     0 SLAAG:       CLK FAAL:
```

```
112       t5 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:        f
113       t5 VEKT:     0 RES:     4 SLAAG:       CLK FAAL:        f
114       t5 VEKT:     0 RES:     0 SLAAG:       CLK FAAL:
115      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
116      CLK VEKT:     4 RES:   246 SLAAG:           FAAL:
117       t6 VEKT:    48 RES:     0 SLAAG:        f0 FAAL:
118       t6 VEKT:     0 RES:     0 SLAAG:        f0 FAAL:
119       t6 VEKT:     0 RES:     0 SLAAG:        f0 FAAL:       f1(
120       t6 VEKT:     0 RES:     8 SLAAG:        f0 FAAL:        f
121      END VEKT:     0 RES:     0 SLAAG:           FAAL:
122       f0 VEKT:     1 RES:     0 SLAAG:           FAAL:
123    FFREE VEKT:     0 RES:     0 SLAAG:           FAAL:
124       f1 VEKT:     1 RES:     0 SLAAG:           FAAL:
125u12-14/0 VEKT:     0 RES:     0 SLAAG: u12-12/1 FAAL:
126       f2 VEKT:     1 RES:     0 SLAAG:           FAAL:
127 u12-7/0 VEKT:     0 RES:     0 SLAAG:  u12-9/1 FAAL: B-CL/dis
128       f3 VEKT:     1 RES:     0 SLAAG:           FAAL:
129u13-14/0 VEKT:     0 RES:     0 SLAAG: u13-12/1 FAAL:
130       f4 VEKT:     1 RES:     0 SLAAG:           FAAL:
131 u13-7/0 VEKT:     0 RES:     0 SLAAG:  u13-9/1 FAAL:
132       f5 VEKT:     2 RES:     0 SLAAG:           FAAL:
133u12-12/0 VEKT:     0 RES:     0 SLAAG: u12-14/1 FAAL: A-CL/dis
134 u15-2/0 VEKT:     0 RES:     0 SLAAG:  u15-2/1 FAAL:
135       f6 VEKT:     1 RES:     0 SLAAG:           FAAL:
136 u12-9/0 VEKT:     0 RES:     0 SLAAG:           FAAL:
137       f7 VEKT:     1 RES:     0 SLAAG:           FAAL:
138u13-12/0 VEKT:     0 RES:     0 SLAAG: C-CL/dis FAAL:
139       f8 VEKT:     1 RES:     0 SLAAG:           FAAL:
140 u13-9/0 VEKT:     0 RES:     0 SLAAG: D-CL/dis FAAL:
141       f9 VEKT:     1 RES:     0 SLAAG:           FAAL:
142 u12-7/1 VEKT:     0 RES:     0 SLAAG:           FAAL:
143      f10 VEKT:     1 RES:     0 SLAAG:           FAAL:
144u13-14/1 VEKT:     0 RES:     0 SLAAG:           FAAL:
145      END VEKT:     0 RES:     0 SLAAG:           FAAL:
146    O_PRT VEKT:   434 RES:     0 SLAAG:           FAAL:
147    I_PRT VEKT:   437 RES:     1 SLAAG:           FAAL:
148      CLK VEKT:   434 RES:     0 SLAAG:           FAAL:
149      CLK VEKT:     1 RES:    48 SLAAG:           FAAL:
150      CLK VEKT:   434 RES:    48 SLAAG:           FAAL:
151      CLK VEKT:     1 RES:   112 SLAAG:           FAAL:
152      CLK VEKT:   432 RES:   247 SLAAG:           FAAL:
153      CLK VEKT:     1 RES:   246 SLAAG:           FAAL:
154       t1 VEKT:   112 RES:     0 SLAAG:       CLK FAAL:        f1
155      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
156      CLK VEKT:     1 RES:   246 SLAAG:           FAAL:
157       t2 VEKT:   112 RES:     0 SLAAG:       CLK FAAL:        f2
158      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
159      CLK VEKT:     1 RES:   246 SLAAG:           FAAL:
160       t3 VEKT:   112 RES:     0 SLAAG:       CLK FAAL:        f3
161      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
162      CLK VEKT:     1 RES:   246 SLAAG:           FAAL:
163       t4 VEKT:   112 RES:     0 SLAAG:       CLK FAAL:        f4
164      CLK VEKT:   434 RES:     0 SLAAG:           FAAL:
165      CLK VEKT:     1 RES:    48 SLAAG:           FAAL:
166      CLK VEKT:   432 RES:   247 SLAAG:           FAAL:
167      CLK VEKT:     1 RES:   246 SLAAG:           FAAL:
```

```
168      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
169      CLK VEKT:     4 RES:   246 SLAAG:          FAAL:
170      CLK VEKT:   434 RES:    48 SLAAG:          FAAL:
171      CLK VEKT:     1 RES:   112 SLAAG:          FAAL:
172       t5 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:          f
173      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
174      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
175       t6 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:          f
176      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
177      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
178       t7 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:          f
179      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
180      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
181       t8 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:          f
182      CLK VEKT:   434 RES:     0 SLAAG:          FAAL:
183      CLK VEKT:     1 RES:    48 SLAAG:          FAAL:
184      CLK VEKT:   432 RES:   247 SLAAG:          FAAL:
185      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
186      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
187      CLK VEKT:     8 RES:   246 SLAAG:          FAAL:
188      CLK VEKT:   434 RES:    48 SLAAG:          FAAL:
189      CLK VEKT:     1 RES:   112 SLAAG:          FAAL:
190       t9 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:          f
191      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
192      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
193      t10 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:         f1
194      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
195      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
196      t11 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:         f1
197      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
198      CLK VEKT:     1 RES:   246 SLAAG:          FAAL;
199      t12 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:         f1
200      CLK VEKT:   434 RES:     0 SLAAG:          FAAL:
201      CLK VEKT:     1 RES:    48 SLAAG:          FAAL:
202      CLK VEKT:   432 RES:   247 SLAAG:          FAAL:
203      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
204      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
205      CLK VEKT:    12 RES:   246 SLAAG:          FAAL:
206      CLK VEKT:   434 RES:    48 SLAAG:          FAAL:
207      CLK VEKT:     1 RES:   112 SLAAG:          FAAL:
208      t13 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:         f1
209      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
210      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
211      t14 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:         f1
212      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
213      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
214      t15 VEKT:   112 RES:     0 SLAAG:      CLK FAAL:         f1
215      CLK VEKT:   432 RES:   254 SLAAG:          FAAL:
216      CLK VEKT:     1 RES:   246 SLAAG:          FAAL:
217      t16 VEKT:   112 RES:     0 SLAAG:       f0 FAAL:         f1
218      END VEKT:     0 RES:     0 SLAAG:          FAAL:
219       f0 VEKT:     1 RES:     0 SLAAG:          FAAL:
220    FFREE VEKT:     0 RES:     0 SLAAG:          FAAL:
221       f1 VEKT:     2 RES:     0 SLAAG:          FAAL:
222   u9-7/1 VEKT:     0 RES:     0 SLAAG:   u9-6/1 FAAL:   u2-10/
223    CL/1 VEKT:     0 RES:     0 SLAAG:          FAAL:
```

```
224       f2 VEKT:     1 RES:     0 SLAAG:           FAAL:
225   u2-5/1 VEKT:     0 RES:     0 SLAAG:    u2-4/0 FAAL:
226       f3 VEKT:     1 RES:     0 SLAAG:           FAAL:
227   u1-9/1 VEKT:     0 RES:     0 SLAAG:   u1-10/0 FAAL:
228       f4 VEKT:     1 RES:     0 SLAAG:           FAAL:
229   u1-5/1 VEKT:     0 RES:     0 SLAAG:    u1-4/0 FAAL:
230       f5 VEKT:     1 RES:     0 SLAAG:           FAAL:
231   u4-9/1 VEKT:     0 RES:     0 SLAAG:   u4-10/0 FAAL:
232       f6 VEKT:     1 RES:     0 SLAAG:           FAAL:
233   u4-5/1 VEKT:     0 RES:     0 SLAAG:    u4-4/0 FAAL:
234       f7 VEKT:     1 RES:     0 SLAAG:           FAAL:
235   u3-9/1 VEKT:     0 RES:     0 SLAAG:   u3-10/0 FAAL:
236       f8 VEKT:     1 RES:     0 SLAAG:           FAAL:
237   u3-5/1 VEKT:     0 RES:     0 SLAAG:    u3-4/0 FAAL:
238       f9 VEKT:     1 RES:     0 SLAAG:           FAAL:
239   u6-9/1 VEKT:     0 RES:     0 SLAAG:   u6-10/0 FAAL:
240      f10 VEKT:     1 RES:     0 SLAAG:           FAAL:
241   u6-5/1 VEKT:     0 RES:     0 SLAAG:    u6-4/0 FAAL:
242      f11 VEKT:     1 RES:     0 SLAAG:           FAAL:
243   u5-9/1 VEKT:     0 RES:     0 SLAAG:   u5-10/0 FAAL:
244      f12 VEKT:     1 RES:     0 SLAAG:           FAAL:
245   u5-5/1 VEKT:     0 RES:     0 SLAAG:    u5-4/0 FAAL:
246      f13 VEKT:     1 RES:     0 SLAAG:           FAAL:
247   u8-9/1 VEKT:     0 RES:     0 SLAAG:   u8-10/0 FAAL:
248      f14 VEKT:     1 RES:     0 SLAAG:           FAAL:
249   u8-5/1 VEKT:     0 RES:     0 SLAAG:    u8-4/0 FAAL:
250      f15 VEKT:     1 RES:     0 SLAAG:           FAAL:
251   u7-9/1 VEKT:     0 RES:     0 SLAAG:   u7-10/0 FAAL:
252      f16 VEKT:     1 RES:     0 SLAAG:           FAAL:
253   u7-5/1 VEKT:     0 RES:     0 SLAAG:    u7-4/0 FAAL:
254      END VEKT:     0 RES:     0 SLAAG:           FAAL:
255    O_PRT VEKT:   434 RES:     0 SLAAG:           FAAL:
256    I_PRT VEKT:   437 RES:     1 SLAAG:           FAAL:
257      CLK VEKT:   434 RES:     0 SLAAG:           FAAL:
258      CLK VEKT:     1 RES:    48 SLAAG:           FAAL:
259      CLK VEKT:   434 RES:    48 SLAAG:           FAAL:
260      CLK VEKT:     1 RES:   112 SLAAG:           FAAL:
261      CLK VEKT:   432 RES:     7 SLAAG:           FAAL:
262      CLK VEKT:     1 RES:     6 SLAAG:           FAAL:
263      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
264      CLK VEKT:     3 RES:   246 SLAAG:           FAAL:
265       t1 VEKT:   112 RES:     1 SLAAG:       CLK FAAL:      f
266      CLK VEKT:   434 RES:     0 SLAAG:           FAAL:
267      CLK VEKT:     1 RES:    48 SLAAG:           FAAL:
268      CLK VEKT:   434 RES:    48 SLAAG:           FAAL:
269      CLK VEKT:     1 RES:   112 SLAAG:           FAAL:
270      CLK VEKT:   432 RES:     7 SLAAG:           FAAL:
271      CLK VEKT:     1 RES:     6 SLAAG:           FAAL:
272      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
273      CLK VEKT:     7 RES:   246 SLAAG:           FAAL:
274       t2 VEKT:   112 RES:     1 SLAAG:       CLK FAAL:      f
275      CLK VEKT:   434 RES:     0 SLAAG:           FAAL:
276      CLK VEKT:     1 RES:    48 SLAAG:           FAAL:
277      CLK VEKT:   434 RES:    48 SLAAG:           FAAL:
278      CLK VEKT:     1 RES:   112 SLAAG:           FAAL:
279      CLK VEKT:   432 RES:     7 SLAAG:           FAAL:
```

```
280      CLK VEKT:     1 RES:     6 SLAAG:           FAAL:
281      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
282      CLK VEKT:    11 RES:   246 SLAAG:           FAAL:
283       t3 VEKT:   112 RES:     1 SLAAG:       CLK FAAL:        f
284      CLK VEKT:   434 RES:     0 SLAAG:           FAAL:
285      CLK VEKT:     1 RES:    48 SLAAG:           FAAL:
286      CLK VEKT:   434 RES:    48 SLAAG:           FAAL:
287      CLK VEKT:     1 RES:   112 SLAAG:           FAAL:
288      CLK VEKT:   432 RES:     7 SLAAG:           FAAL:
289      CLK VEKT:     1 RES:     6 SLAAG:           FAAL:
290      CLK VEKT:   432 RES:   254 SLAAG:           FAAL:
291      CLK VEKT:    15 RES:   246 SLAAG:           FAAL:
292       t4 VEKT:   112 RES:     1 SLAAG:        f0 FAAL:        f
293      END VEKT:     0 RES:     0 SLAAG:           FAAL:
294       f0 VEKT:     1 RES:     0 SLAAG:           FAAL:
295    FFREE VEKT:     0 RES:     0 SLAAG:           FAAL:
296       f1 VEKT:     3 RES:     0 SLAAG:           FAAL:
297  u1-5/0 VEKT:     0 RES:     0 SLAAG:   u1-9/0 FAAL:    u2-5/
298  u2-9/0 VEKT:     0 RES:     0 SLAAG:   u1-4/1 FAAL:      CL/
299  u9-7/0 VEKT:     0 RES:     0 SLAAG:           FAAL:
300       f2 VEKT:     2 RES:     0 SLAAG:           FAAL:
301  u3-5/0 VEKT:     0 RES:     0 SLAAG:   u3-9/0 FAAL:    u4-5/
302  u4-9/0 VEKT:     0 RES:     0 SLAAG:   u3-4/1 FAAL:
303       f3 VEKT:     2 RES:     0 SLAAG:           FAAL:
304  u5-5/0 VEKT:     0 RES:     0 SLAAG:   u5-9/0 FAAL:    u6-5/
305  u6-9/0 VEKT:     0 RES:     0 SLAAG:   u5-4/1 FAAL:
306       f4 VEKT:     2 RES:     0 SLAAG:           FAAL:
307  u7-5/0 VEKT:     0 RES:     0 SLAAG:   u7-9/0 FAAL:    u8-5/
308  u8-9/0 VEKT:     0 RES:     0 SLAAG:   u7-4/1 FAAL:   u13-7/
309      END VEKT:     0 RES:     0 SLAAG:           FAAL:
310      END VEKT:     0 RES:     0 SLAAG:           FAAL:
```

Central University of Technology, Free State

G

Resultaatlêer

Vektor:	6	Resultaat:	0	Uitset:	0
Vektor:	0	Resultaat:	0	Uitset:	0
Vektor:	6	Resultaat:	0	Uitset:	0
Vektor:	32	Resultaat:	64	Uitset:	64
Vektor:	6	Resultaat:	0	Uitset:	0
Vektor:	0	Resultaat:	0	Uitset:	0
Vektor:	14	Resultaat:	0	Uitset:	0
Vektor:	0	Resultaat:	0	Uitset:	0
Vektor:	10	Resultaat:	32	Uitset:	32
Vektor:	0	Resultaat:	16	Uitset:	16
Vektor:	14	Resultaat:	32	Uitset:	32
Vektor:	0	Resultaat:	16	Uitset:	16
Vektor:	15	Resultaat:	32	Uitset:	32
Vektor:	0	Resultaat:	16	Uitset:	16
Vektor:	7	Resultaat:	0	Uitset:	0
Vektor:	0	Resultaat:	0	Uitset:	0
Vektor:	0	Resultaat:	0	Uitset:	0
Vektor:	48	Resultaat:	0	Uitset:	0
Vektor:	32	Resultaat:	0	Uitset:	0
Vektor:	32	Resultaat:	0	Uitset:	0
Vektor:	48	Resultaat:	128	Uitset:	0

LITERATUURLYS

Anoniem, 1990. The PC instrument bus. Electronics World + Wireless World. April 1990, bl.309-312.

Eagle Electric, 1991. Eagle Electronic Catalogue. Kaapstad: Eagle Electric.

Capillo, C. 1990. Surface mount technology: Materials, processes, and equipment. New York: McGraw-Hill.

Fluke. 1987: Digital test system. Pulse/SA Measurement & Control. September 1987, bl.18-20.

Fujiwara, H. 1990. Computational Complexity of Controllability/Observability Problems for Combinational Circuits. IEEE Transactions on Computers, vol.39, nr.6, Junie 1990, bl.762-767.

Fujiwara, H. en Shimono, T. 1983. On the Acceleration of Test Generation Algorithms. IEEE Transactions on Computers, vol.c-32, nr.12, Desember 1983, bl.1137-1144.

Ginsberg, G.L. 1991. Printed circuit design: Featuring computer-aided technologies. New York: McGraw-Hill.

Hakim, B. 1989. Microelectronic Reliability: Reliability, Test and Diagnostics. Norwood: Artech House.

Healy, J.T. 1981. Automatic Testing and Evaluation of Digital Integrated Circuits. Virginia: Prentice-Hall.

Herington, D.E, Nichols, P.A. en Lipp, R.D. 1987. Software Verification Using Branch Analysis. Hewlett-Packard Journal, vol. 38, nr. 6, Junie 1987, bl.13-22.

Hinch, S.W. 1988. Handbook of Surface Mount Technology. U.K.: Longman Group.

Kodandapani, K.L. en Pradhan, D.K. 1980. Undetectability of Bridging Faults and Validity of Stuck-At Fault Test Sets. IEEE Transactions on Computers, vol.c-29, nr.1, Januarie 1980, bl.55-59.

Loveday, G.C. 1980. Electronic Testing and Fault Diagnosis. London: Pitman Education Limited.

McCluskey, E.J. en Bozorgui-Nesbat, S. 1981. Design for Autonomous Test. IEEE Transactions on Computers, vol.c-30, nr.11, November 1981, bl.866-875.

Morant, M.J. 1990. Integrated Circuit Design and Technology. London: Chapman and Hall.

Nazili, K. 1988. What to expect from a low cost tester. Pulse/SA Measurement & Control, Mei 1988, bl.23-24.

Pitchumani, V en Soman, S.S. 1988. Functional Test Generation Based on Unate Function Theory. IEEE Transactions on Computers, vol.37, nr.6, Junie 1988, bl.756-760.

Pynn, C. 1986. Strategies for Electronics Test. New York: McGraw-Hill.

Randell, B en Treleaven, P.C. 1983. VLSI architecture. New York: Prentice-Hall.

Seth, S.C, Agrawal, V.D. en Farhat, H. 1990. A statistical Theory of Digital Circuit Testability. IEEE Transactions on Computers, vol.39, nr.4, April 1990, bl.582-586.

Spencer, T.H. en Savir, J. 1985. Layout Influences Testability. IEEE Transactions on Computers, vol.c-34, nr.3, Maart 1985, bl.287-290.

Sutton, J.C. en Bredeson, J.G. 1980. Minimal Redundant Logic for High Reliability and Irredundant Testability. IEEE Transactions on Computers, vol.c-29, nr.7, Julie 1980, bl.648-655.

Turino, J. 1990. Design to Test. A definitive guide for Electronic Design, Manufacture, and Service. 2de uitgawe, New York: Van Nostrand Reinhold

Wilkens, B.R. 1986. Testing Digital Circuits: an Introduction. England: Van Nostrand Reinhold

Wilkenson, B. 1987. Digital System Design. London: Prentice-Hall

Yarmolik, V.N. 1990. Fault Diagnosis of Digital Circuits. Chichester: John Wiley & Sons.

Wilkenson, B. 1987. Digital System Design. London: Prentice-Hall

Yarmolik, V.N. 1990. Fault Diagnosis of Digital Circuits. Chichester: John Wiley & Sons.