

TESTNO USMERJEN JEZIK TESTOL

J. ŠILC,
P. KOLBEZEN

UDK: 681.3.06

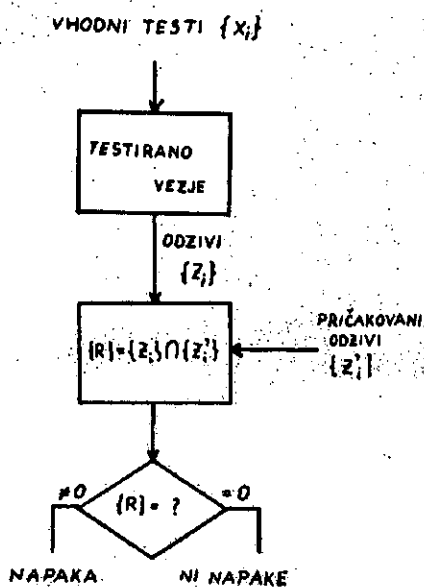
INSTITUT JOŽEF STEFAN, LJUBLJANA

V delu je podan pregleden opis programskega jezika TESTOL, ki omogoča bolj ali manj učinkovito statično, dinamično in funkcionalno testiranje digitalnih vezij in sistemov.

TEST ORIENTED LANGUAGE TESTOL. In this paper a review of programming language TESTOL is presented which enables efficient static, dynamic and functional testing of digital circuits and systems.

I. UVOD

Testiranje digitalnih vezij se v splošnem sestoji iz zaporednega dodajanja testnih naborov X_i , $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, na vhode testiranega vezja in opazovanja ustreznih odzivov Z_i , $Z_i = (z_{i1}, z_{i2}, \dots, z_{im})$ ter primerjanje le-teh z že vnaprej izračunanimi oz. pričakovanimi odzivi Z'_i . Vsako protislovje oz. neujemanje določa napako (slika 1.).



Slika 1.

Takšno testiranje je izvedeno s pomočjo mini ali mikro računalnika in ustreznega vmesnika,

ki omogoča komuniciranje računalnika s testiranim vezjem (slika 2.). Vmesnik nam omogoča hkratno vpisovanje oz. branje z vseh vhodnih oz. izhodnih priključkov testiranega vezja.



Slika 2.

Predno začnemo s testiranjem, je potrebno podati računalniku informacijo o testiranem vezju. Definirati moramo vhodne in izhodne priključke testiranega vezja, določiti obliko in časovni potek vhodnih testov X_i in pričakovane odzive Z'_i . V ta namen je bil determiniran uporabniško usmerjen jezik TESTOL (TEST Oriented Language).

2. TESTNO USMERJEN VHODNI JEZIK TESTOL

TESTOL nam omogoča enostaven in učinkovit opis vhodnih in izhodnih priključkov testiranega vezja, kakor tudi enostavno določitev vhodnih testov in ustreznih pričakovanih odzivov. S pomočjo TESTOL-a je mogoče določiti:

- vhodne priključke,
- izhodne priključke,
- inicializacijske vhodne nabore,
- vhodne teste,
- pričakovane odzive in
- časovni potek testiranja, to je zakasnitve

med posameznimi vhodnimi testi, kakor tudi zakasnitve med testi in ustreznimi odzivi.

Sintaksa TESTOL-a je opisana s sintaktičnimi diagrami. Program napisan v TESTOL-u je sestavljen iz dveh delov, ki ju imenujemo "glava programa" in "blok" (slika 3.).

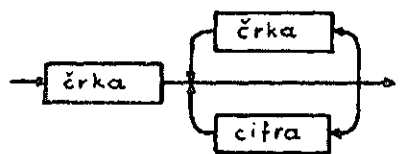


Slika 3. Program

Glava programa ima to nalogo, da da testnemu programu ime. Blok je sestavljen iz petih stavkov, ki si morajo slediti v predpisanem vrstnem redu, kot je prikazano na sliki 4. Pripomniti velja, da lahko presledke in prehode v novo vrstico uporabljamo kjerkoli v programu, pač glede na preglednost programa.

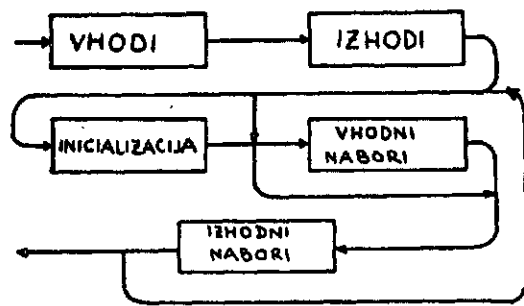
STAVEK VHODI

Vhodne priključke testiranega vezja opišemo s stavkom VHODI. Množici priključkov (oznake priključkov so identične oznakam na evropa konektorju, preko katerega je testirano vezje priključeno na vmesnik) lahko priredimo simbolično ime, ki ima lahko največ 10 alfanumeričnih znakov in se začneja s črko (slika 5.). Število



Slika 5. Ime

priključkov, ki pripadajo imenu ni omejeno. Pri navajanju priključkov je pomembno, da si sledijo v pravem vrstnem redu. Ta podrobnost je pomembna zato, ker pri opisovanju vhodnih oz. inicializacijskih naborov prirejamo posameznim imenom določene vrednosti in je s pozicijo določenega priključka v imenu, določena njegova binarna vrednost (slika 6.). Pri navajanju priključkov na vezja realizirana na tiskanini dvojnega evropa formata, moramo navesti kateremu konektorju priključki pripadajo (slika 7.). Če oznake konektorja ne navedemo se smatra, da gre za konektor K1. Sintaksa stavka VHODI je prikazana na sliki 8.



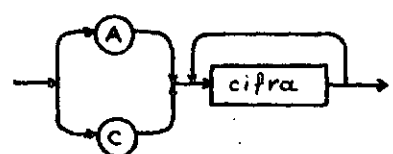
Slika 4. Blok

STAVEK IZHODI

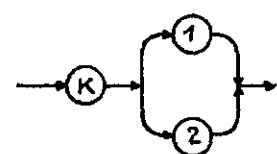
S stavkom IZHODI opišemo izhodne priključke testiranega vezja. Enako kot v stavku VHODI tudi tu množici priključkov priredimo simbolično ime. Vrstni red navedenih priključkov ima enak smisel kot pri navajanju vhodnih priključkov. Velja pripomniti, da isti priključek ne more obenem pripadati množici vhodnih in množici izhodnih priključkov. Sintaksa stavka IZHODI je prikazana na sliki 9.

STAVEK VHODNI NABORI

S stavkom VHODNI NABORI določimo število, obliko in časovni potek vhodnih testov, kakor tudi informacijo o tem, ali in po kolikem času opa-



Slika 6. Oznaka priključka



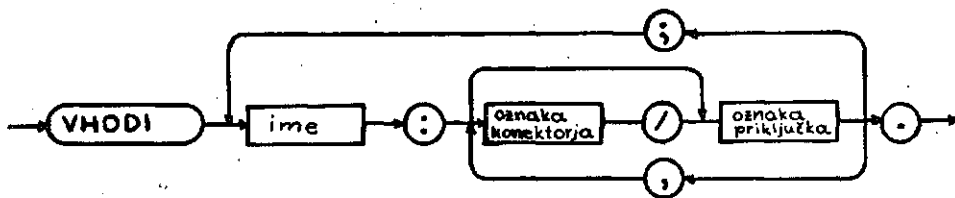
Slika 7. Oznaka konektorja

zujemo ustrezne odzive. Število vhodnih testov je podano s parametrom "število korakov" (slika 10.). Z enim stavkom VHODNI NABORI definiramo m, $1 \leq m \leq 1030$, vhodnih testov. Številu korakov sledijo imena, ki so določena v stavku VHODI, katerim priredimo določene vrednosti, ki jih navajamo v desetiškem številskem sestavu. Imenom, ki jih eksplicitno ne navedemo, se priredi vrednost 0. Vrednosti imen se pretvorijo v dvojiški številski sestav in posamezni biti

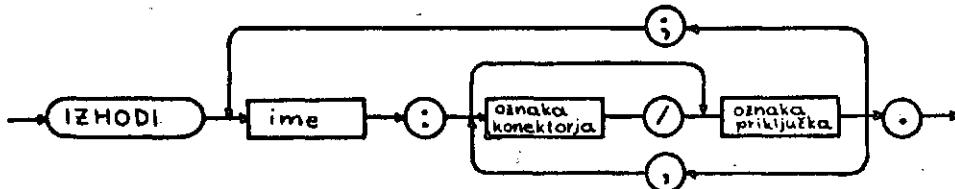


Slika 10. Število korakov

mo m, $1 \leq m \leq 1030$, vhodnih testov. Številu korakov sledijo imena, ki so določena v stavku VHODI, katerim priredimo določene vrednosti, ki jih navajamo v desetiškem številskem sestavu. Imenom, ki jih eksplicitno ne navedemo, se priredi vrednost 0. Vrednosti imen se pretvorijo v dvojiški številski sestav in posamezni biti



Slika 8. Stavak VHODI

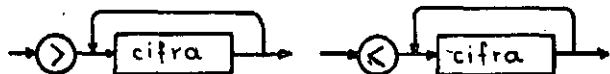


Slika 9. Stavak IZHODI

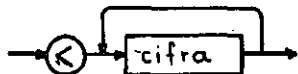
predstavljajo, v skladu z vrstnim redom priključkov v sklopu imena, binarno vrednost vhodnega priključka. V vsakem koraku se nad vrednostjo imena lahko izvrši ena od naslednjih aritmetičnih operacij:

1. INC (inkrement) - vrednosti imena se v vsakem koraku prišteje 1
2. DEC (dekrement) - vrednosti imena se v vsakem koraku odšteje 1
3. NEG (negacija) - vrednost imena se v vsakem koraku negira
4. SUB n (odštevanje) - vrednosti imena se v vsakem koraku odšteje n
5. ADD n (prištevanje) - k vrednosti imena se v vsakem koraku prišteje n
6. MUL n (množenje) - vrednost imena se v vsakem koraku množi z n
7. DIV n (deljenje) - vrednost imena se v vsakem koraku integer delji z n .

Sintaksa aritmetične operacije je podana na sliki 11. Nadalje je s stavkom VHODNI NABORI možno podati časovni potek vhodnih testov. To nam omogoča parameter "vhodna zakasnitev". (slika 12.). Za znakom \gg navedemo faktor vhodne zakasnitve N , ki nam pove, kolikokrat bo T_{vn} (čas med vhodnimi nabori) večji od minimalne zakasnitve τ ; $T_{vn} = N\tau$. Minimalna zakasnitev τ je



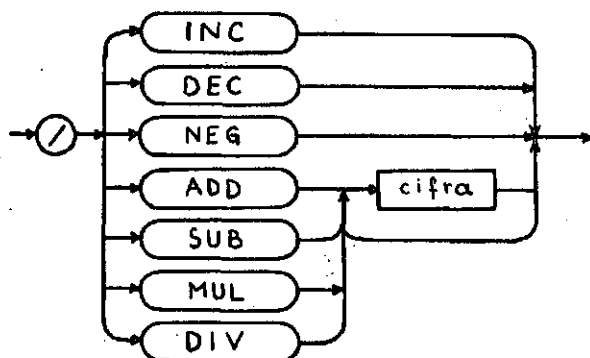
Slika 12. Vhodna zakasnitev



Slika 13. Izhodna zakasnitev

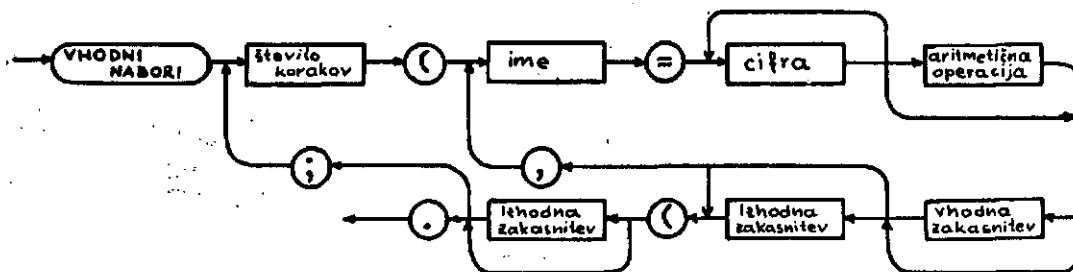
odvisna od računalnika, ki testiranje krmili, maksimalni T_{vn} je omejen z maksimalnim celim številom N_{max} , ki ga ta računalnik lahko procesira; $1 \leq N \leq N_{max}$. Če parameter "vhodna zakasnitev" izpustimo velja $N = 1$. Stavak VHODNI NABORI nam omogoča določiti tudi časovni potek opazovanja odzivov vezja. To nam omogoča parameter "izhodna zakasnitev" (slika 13.). Za znakom \ll navedemo faktor izhodne zakasnitve M , ki

nam pove, kolikokrat bo T_{izh} (čas med vhodnim testom in opazovanjem ustreznega odziva) večji od minimalne zakasnitve τ_0 ; $T_{izh} = M\tau_0$. Parameter je mogoče pisati pred ali za okrogli zaklepaj (slika 14.). V prvem primeru, ko je izhodna zakasnitev znotraj okroglega oklepaja, se izhodna zakasnitev nanaša na m odzivov, ki pripadajo m vhodnim testom. Takšna uporaba je smiselna le v primeru, ko je $T_{izh} \leq T_{vn}$. V drugem primeru, ko je izhodna zakasnitev podana za okroglim zaklepajem, opazujemo odziv testirane



Slika 11. Aritmetična operacija

ga vezja, ki je posledica vhodnega testa oz. množice vhodnih testov, po času T_{izh} za zadnjim vhodnim testom. Če parametra "izhodna zakasnitev" ne pišemo, to ne pomeni, da je $M = 1$ (to moramo eksplicitno napisati), temveč, da je čas T_{izh} neskončno oz., da ustreznega odziva ne opazujemo. Sintaksa stavka VHODNI NABORI je prikazana na sliki 14.



Slika 14. Stavek VHODNI NABORI

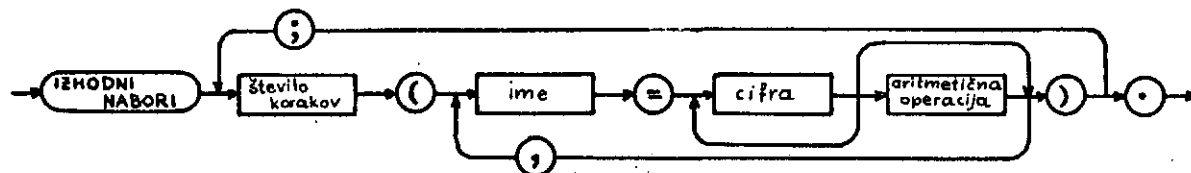
STAVEK IZHODNI NABORI

S stavkom IZHODNI NABORI definiramo pričakovane odzive testiranega vezja. Pri navajanju pričakovanih odzivov moramo paziti, da se njihovo zaporedje ujema z zaporedjem ustreznih vhodnih testov. Tudi tu velja, da se imenom, ki jih eksplicitno ne navedemo priredi vrednost 0. Sintaksa stavka IZHODNI NABORI je prikazana na sliki 15.

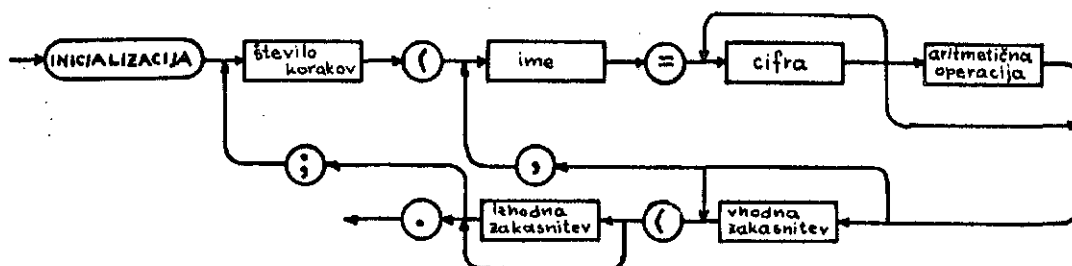
čakovan odziv testiranega vezja na inicializacijske nabore. V primeru, ko se dejanski in pričakovani odziv razlikujeta inicializacijo ponovimo.

3. ZAKLJUČEK

Namen pričujočega članka je predvsem seznaniti bralca z osnovnimi značilnostmi testno usmerjenega jezika TESTOL-a. S pomočjo jezika TESTOL



Slika 15. Stavek IZHODNI NABORI



Slika 16. Stavek INICIALIZACIJA

STAVEK INICIALIZACIJA

Sekvenčna vezja je pred pričetkom testiranja koristno inicializirati, saj se po priključitvi na napajalno napetost postavijo v nedefinirano stanje. Če sekvenčna vezja nimajo posebnih sinhronizacijskih vhodov, jih je potrebno s pomočjo normalnih vhodnih signalov, ki jih določimo s stavkom INICIALIZACIJA, postaviti v znano stanje. Stavek INICIALIZACIJA (slika 16.) je sintaktično podoben stavku VHODNI NABORI. Z njim generiramo inicializacijski nabor oz. množico inicializacijskih naborov, ki nam postavi sekvenčno vezje v znano stanje, enako kot vhodne teste. Stavku INICIALIZACIJA lahko sledi stavku IZHODNI NABORI, s katerim določimo pri-

je mogoče izvajati statično, dinamično in (predvsem) funkcionalno testiranje. Prevajalnik za jezik TESTOL je napisan v visokem programskem jeziku PASCAL.

4. LITERATURA

- (1) A. KRIŽNIK: Diplomaska naloga, Ljubljana 1979
- (2) K. JENSEN, N. WIRTH: PASCAL. (User Manual and Report) Springer - Verlag, New-York Heidelberg Berlin, 1974