

Testovanie mikroprocesorov

Problémy: nedostatok informácií o vnútornej štruktúre – pozná ju iba výrobca, katalógové schémy treba považovať iba za náhradné - čierna skrinka, šedá skrinka.

Požiadavky:

potreba znalosti: štruktúry - náhradné katalógové schémy
programovacieho jazyka

Možnosti riešenia:

- aktivizácia blokov – overenie ich funkcie

testovanie jadra

postupné nabaľovanie ďalších blokov

- aplikačné programy –
Výhody: malé náklady na diagnostiku
Nevýhody: nezaručujú úplnosť testu
potreba simulovať okolie: vstupy a
očakávané výstupy
- pseudonáhodná postupnosť inštrukcií s vonkajším skúšačom
Výhody: jednoduché generovanie
Nevýhody: kontrola prípustnosti inštrukčného kódu
kontrola úplnosti testu
- autonómne testy
- vzájomné skúšanie dvoch MP
- Použitie analyzátorov: logických
príznakových

lokalizácia porúch vo vnútri IO, degradácia funkcie

Generovanie testov mikroprocesorov.

Nízka efektívnosť známych metód generovania testov LO v dôsledku:

- veľkého počtu prvkov v štruktúre μP
- veľkého počtu väzieb v štruktúre μP
- obmedzeného počtu vývodov z puzdra IO
- neznalosť presnej štruktúry obvodu

Prístupy:

- testovanie μP prostredníctvom testov jeho modulov so známou resp. neznámou štruktúrou
- testovanie ucelených operácií v blokoch ČP
- testovanie čiastkových operácií v blokoch ČP

- a) blok spracovania údajov: aritmetické a logické operácie
modifikácie operandov a výsledkov
spracovania príznakov
výpočet adries
- b) blok riadenia spracovania údajov:
 - dekodovanie operácie a ich modifikácii
 - aktivizácia operácií
- c) blok uchovania a prenosu údajov
- d) blok riadenia uchovania a prenosu údajov
 - výber registrov
 - adresácia
 - reakcia na vnútorné stavy
- e) blok reakcie na vonkajšie signály a operácie vstupu výstupu

Test mikroprocesora - test každého jeho bloku (mechanizmu) za predpokladu, že ostatne bloky fungujú správne.

Testovanie bloku spracovania údajov

Využíva sa: **znalosť výsledkov** operácií nad známymi operandami
regulárnosť štruktúry spočívajúca v zákonitom opakovaní prvku štruktúry a väzieb medzi nimi

Logické operácie - operácie nad jednotlivými bitmi príslušných operandov

- výpočet po bitoch dvoch resp. jedného operandu
- pri operáciách posuvu sa uvažuje aj prenos zo susedného rádu
- blok vykonávania logických operácií predstavuje regulárnu štruktúru s n zložkami, kde n je počet bitov operandov
- test pre ľubovoľný súbor porúch pozostáva zo štyroch (alebo aj dvoch) krokov (operandov)

Operácie sčítania - so sériovým prenosom – iteračná štruktúra minimálny test obsahuje 8 krokov

- s paralelným resp. skupinovým prenosom vedie na zložitejšie testy je možné využiť hierarchiu štruktúry na rekurzívnu tvorbu testov

Operácie modifikácie operandov

- štruktúra je daná funkciami
- testy využívajú znalosti osobitosti štruktúry

ad b – funkčný model diagnostiky bloku riadenia spracovania údajov môže byť vyjadrený hodnotou z_i , i -teho bitu výst. registra

$$z_i = \bigcup_{j=1}^m u_i^j \cdot f_i^j + z_i^* \cdot \prod_{j=1}^m \bar{u}_i^j \quad i=1 \text{ až } n$$

kde

u_i^j – signál aktivujúci operáciu vyvolanú inštrukciou I_j v i -tom rade

f_i^j – hodnota i -tého bytu pri inštrukcii I_j

z_i^* – hodnota z_i pred vykonaním operácie

m – počet inštrukcií I

Model porúch:

- niektoré premenné u_i^j nadobúdajú trvalú konštantnú hodnotu 0 resp. 1 a tým namiesto operácie I_j sa vykonáva ľubovoľná podmnožina operácií I_k včítane prázdnej množiny

Nutná a postačujúca podmienka testovania:

pre každú operáciu I_j test obsahuje príkazy:

- a) s operáciou I_j a operandami zabezpečujúcimi rozdielne hodnoty funkcií f_i a premennej z_i^* aspoň pre jeden bit spracovávaných údajov

- b) s operáciou $I_k \neq I_j$ a operandami zabezpečujúcimi, ak je to možné, hodnoty funkcií $f_i^j = 1$ a $f_i^k = 0$

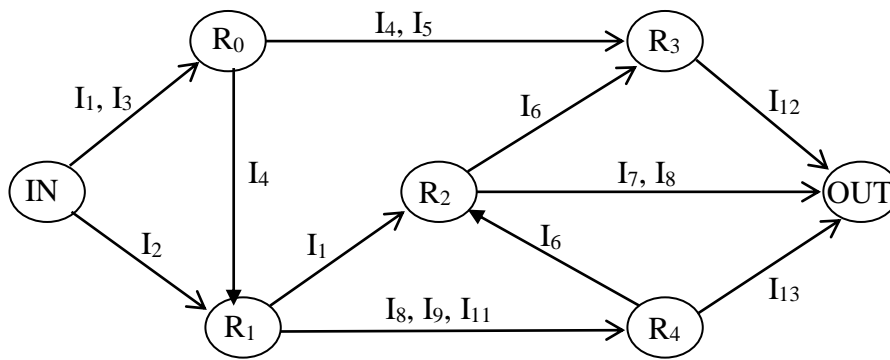
c) Diagnostický model prenosu a uchovania údajov

Údaje sú uchovávané v registroch prenášajú sa medzi registrami priamo resp. prostred. kombinačnej logiky.

Uchovanie a prenos údajov sa môže vyjadriť prostredníctvom grafu registrových prenosov (GRP).

Vrcholy grafu zodpovedajú registrom R_0, R_1, \dots, R_{m-1} a vstupu IN a výstupu OUT.

Hrany grafu s vrcholmi i do j zodpovedajú prenosom $R_i \rightarrow R_j$ a sú označené zoznamom operácií I_k z množiny prenosových operácií. Ak je niektorá operácia prenosu podmienená, pripisuje sa k tejto operácii do zátvoriek príslušná podmienka.



Obr. 6.2 Príklad grafu registrovaných prenosov

Model porúch:

- 1.) Ľubovoľný bit registra nadobúda t_0 resp. t_1
- 2.) Ľubovoľné prenosové bity nadobúdajú t_0 resp. t_1
- 3.) Ľubovoľné prenosové bity sú skratované
- 4.) Vyskytuje sa ľubovoľný počet uvedených porúch
- predpokladá sa, že riadenie prenosu (vyber zdrojov a liniek prenosu) je správne.

Generovanie testu

Test prenosu $R_i \rightarrow R_j$ musí vyhovovať nasledujúcim požiadavkám

- a) každej dvojici bitov musí byť priradená rozdielna hodnota
- b) pre každý test musia existovať minimálne dva operandy s rozdielnou hodnotou signálu v tomto bite

Pre n -bitové registre minimálna testovacia postupnosť pre splnenie požiadavky

- a) obsahuje $\lceil \log_2 n \rceil$ operandov, v ktorej každý operand získame zámienou polovice hodnôt predchádzajúceho operandu na opačnú hodnotu.

pre plnenie požiadavky b) potrebujeme pridať ešte jeden operand (napríklad 4)

Celková dĺžka minimálneho testu je $\log_2 n + 1$.

- $n=8$
- a)
 - 1) 0000 1111
 - 2) 0011 0011
 - 3) 0101 0101
 - b)
 - 4) 1111 0000

- 1) Aby sme otestovali všetky prenosy, musíme v GRP vytvoriť súbor ciest z vrcholu IN do vrcholu OUT, ktorý pokrýva všetky hrany GRP.

- problém pokrytia hrán súborov ciest z IN do OUT

- 2) Pre každú cestu treba zostrojiť minimálnu postupnosť príkazov, ktorá aktivizuje uvedenú cestu

d) Riadenie prenosu údajov, výber registrov - analógia s riadením spracovania údajov

- kontrola mechanizmu adresácie a podmienených i nepodmienených skokov sa zabezpečuje kontrolou správnej postupnosti prenosov pre jednotlivé typy.

Pri generovaní testov mikroprocesorov môžeme štruktúrne a funkčné testy vhodne kombinovať.

Realizácia testov μ P:

- prostredníctvom vonkajšieho skúšača
- pomocou autonómnych testov

Autonómny test je špeciálny program uložený vo vykonávateľnom tvare v nedeštruktívnych pamätiach testovaného systému, ktorý testovanie riadi aj vyhodnocuje. Predpokladá sa funkčnosť tzv. tvrdého jadra systému.

Alternatívou autonómneho testu je vzájomný test dvoch μ P.

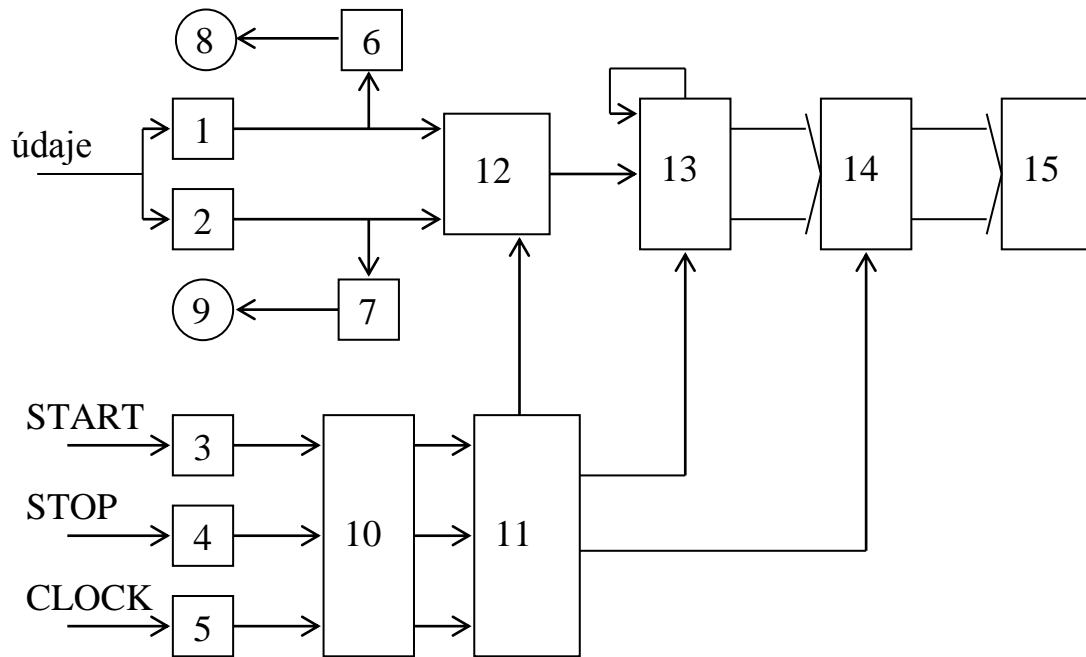
Logický analyzátor

schopnosť zaznamenať a zobrazit' postupnosť hodnôt logických signálov na zbernici pred určitou udalosťou resp. po nej
vyžaduje odbornú obsluhu

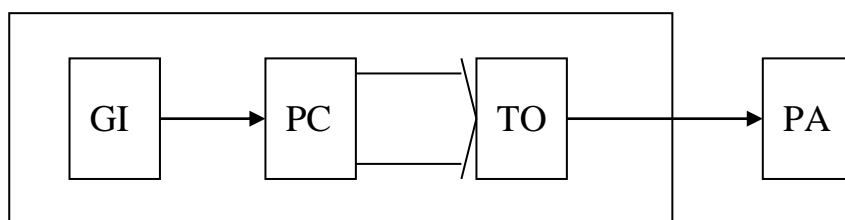
Príznakový analyzátor

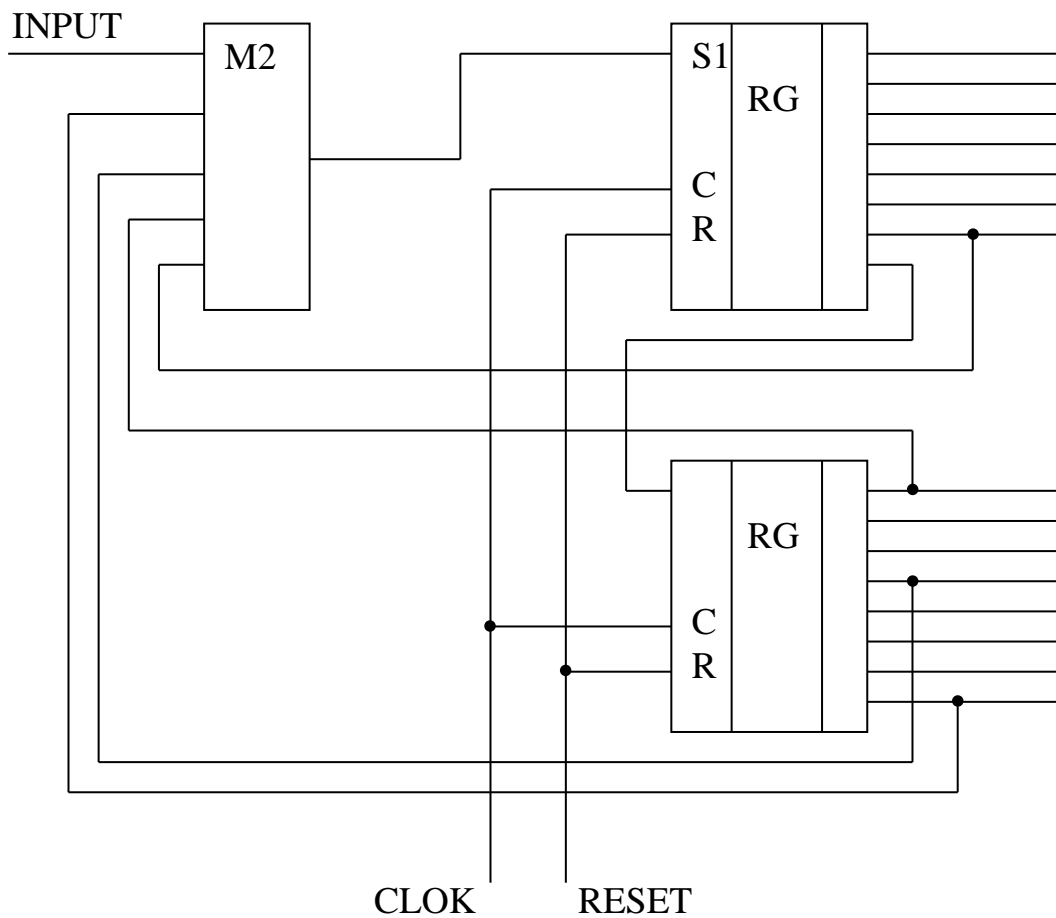
nenáročný a účinný pomocník
vhodný pre synchronne obvody
využíva kompresiu odozvy na 16-bitový príznak

Princíp činnosti príznakového analyzátora.



- 1 – 5 - detektor úrovně (0,8V, 2V, 3 x 1,5V)
- 6 – 7 - predĺženie impulzu
- 8 – 9 - LED indikátor vstupnej úrovně
- 10 - voľba aktívnej úrovně
- 11 - riadenie hradlovania
- 12 - vstupná pamäť údajov
- 13 - generátor cyklického kódu
- 14 - pamäť kódu
- 15 - displej hexadecimálneho kódu





Princíp činnosti generátora cyklického kódu

Dve fázy práce s PA

1. príprava príznakov
2. detekcia a lokalizácia poruchy s využitím SP

Má vysokú detekčnú schopnosť – až 99,99%.

Pri lokalizácii porúch v prípade, že existujú synonyma umiestňujeme signálovú sondu proti smeru postupu signálu, kým nezískame správny príznak a tým lokalizujeme poruchu.