VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Semestrální práce

Moderní teoretická informatika

Využití teorie jazyků v diagnostice obvodu

2002 **Pavel Tupec**

Abstrakt

Formální přístup k problematice testovatelnosti umožňuje formální popis vlastností testovaného obvodu. Tento jednoznačný popis umožňuje transformovat problém analýzy testovatelnosti na již známe a řešené problémy diskrétní matiky a teoretické informatiky. Předmětem popisu je datová část číslicového obvodu na úrovni abstrakce odpovídající úrovni meziregistrových přenosů (Register Transfer - RT). Pro přenos diagnostických dat je zvolen koncept tzv. *i cest*, tedy datových cest, které jsou pro přenášená data transparentní. Předmětem práce je vytvořit a zkoumat formální model jazyka všech *i cest* v analyzovaném obvodu.

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně pod vedením doc. RNDr. Alexandera Meduny, CSc v rámci kurzu TID. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Pavel Tupec

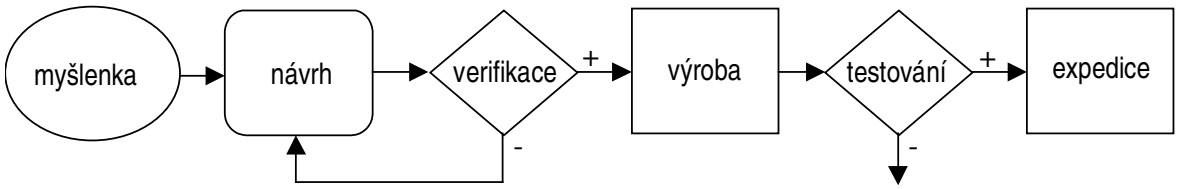
Obsah

Úvod do diagnostiky a testování 3
Formální model obvodu 4
Využití struktury obvodu k přenosu diagnostické informace 7
I cesty 8
Nastavení *i režimu* 10
Shrnutí 12
Literatura 13

Úvod do diagnostiky a testování

Důležitým krokem při výrobě elektronických součástek je posouzení, zda je vyrobený obvod funkční. Pro tyto účely musejí být pro obvod k dispozici diagnostické prostředky, které najdou uplatnění nejen v etapě výroby takového prvku, ale také při jeho použití ve větším elektronickém celku. Důležitým aspektem kvalitně provedeného návrhu je testovatelnost navrženého obvodu. U obvodů menší složitosti byla praxe taková, že principy testovatelnosti byly převážně uplatňovány až po ukončení etapy návrhu obvodu a bylo přesto možné dosáhnout uspokojivých výsledků. Struktura obvodu dovolovala, aby na vstupy testované komponenty bylo možno vkládat testovací vektory a odezvy na ně snímat na výstupu. Tyto metody, kdy je součástka nejprve navržena, zkonstruována a pak jsou teprve realizovány úvahy o testování jsou v diagnostice označovány jako metody ad hoc a jsou dnes, kdy se pracuje s obvody vysoké integrace zcela, nepoužitelné. Při dnešní složitosti navrhovaných obvodů je nutno již při jejich návrhu zvažovat, jak bude obvod testován. V této souvislosti pak mluvíme o testovatelnosti navrhovaného obvodu. Proto vstupují do hry metody označované jako návrh či syntéza obvodu pro snadnou testovatelnost. Syntéza obvodu je proces, při němž je obvod navrhován pomocí CAD nástrojů tak, aby plnil požadovanou funkci a přitom vyhovoval různým kriteriím (velikost zabrané plochy čipu, rychlost, ...).
Z hlediska diagnostiky může být jedním z těchto kriterií i testovatelnost obvodu.

Nefunkčnost obvodu může být způsobena chybou při návrhu nebo závadou vzniklou při výrobě. Disciplína zabývající se touto problematikou je všeobecně známá jako *diagnostika*. Celý cyklus vzniku obvodu s úlohou jednotlivých fází dobře ilustruje obrázek 1.1.



Obr. 1.1: Cyklus vzniku obvodu.

I když bude při verifikaci obvodu zjištěno, že navržený obvod plní správně požadované funkce, není tím zajištěno, že součástka vyrobená podle verifikovaného návrhu bude pracovat bezchybně. V etapě výroby mohou být do obvodu zaneseny poruchy, jejichž projevem bude chybná (jiná než žádaná) funkce obvodu. Navíc se taková chybná funkce obvodu v praxi nemusí ihned projevit, jestliže je zasažena taková funkce obvodu, která není zrovna využívána. Je tedy třeba vyvinout prostředky, které umožní navržený obvod po jeho vyrobení otestovat.

Při úvahách o diagnostice číslicového obvodu je třeba řešit dva okruhy problémů:

- Vytvoření souboru testovacích dat – binárních vektorů, které se budou aplikovat na vstupy testovaných prvků a následně sledovat odezvy prvků na ně tak, aby bylo možno posoudit, zda prvek pracuje správně či nikoliv.
- Způsob aplikace testu – jakým způsobem dopravit testovací data na vstupy prvků, pro něž jsou určeny a jakým způsobem dopravit odezvy na ně od výstupů testovaného prvku k zařízení, které odezvy posuzuje.

Tato práce se pokouší uplatnit teorie jazyků při formulování problematiky aplikace testu na požadované prvky obvodu. Test (soubor testovacích dat) je možné vytvořit ručně nebo programovými prostředky zvanými generátory testu. Generátory testu se obecně rozlišují podle toho, zda generují test pouze kombinačního nebo i sekvenčního obvodu. Pak se hovoří o generátorech testu kombinačních obvodů (GTKO) a generátorech testu sekvenčních obvodů (GTSO). Generování testu pro sekvenční obvody je problém daleko složitější, neboť výstup sekvenčního obvodu závisí kromě vstupů i na vnitřním stavu.

Převzaté definice a věty jsou převzaty s literatury [Ruz02], pro jednoduchost zachovávám jejich číslování v kapitolách a neuvádím důkazy.

Formální model obvodu

Pro jednoznačnost a korektnost dalšího zápisu zde uvádím formální model struktury číslicového obvodu na úrovni RT využívajícího multiplexovaných datových cest zavedeny v literatuře [Ruz02]. Základní pětice množin vyjadřuje staticky strukturu obvodu, k tomu se přidávají zobrazení určující chování prvků a další vztahy. Tento model je navržen tak aby co nejlépe vystihnul strukturu popisu skutečných obvodů tak, jak se v praxi používá např. formou jazyků pro popis obvodů.

Definice 3.1:
Uspořádaná pětice ***UUA=(E, P, C , PI, PO)*** vyjadřuje model struktury číslicového obvodu na úrovni meziregistrových přenosů (RT). Všechny množiny jsou konečné. Význam jednotlivých množin je tento:

E je množina obvodových prvků,
P je množina jejich bran (vstupů a výstupů),
C je množina spojů mezi branami prvků obvodu,
PI je množina primárních vstupů obvodu a
PO je množina primárních výstupů obvodu.

Tento model dobře vyjadřuje hierarchii celého systému, kterého je testovaný obvod ***UUA*** součástí. Vždyť i na celý testovaný obvod ***UUA*** lze pohlížet jako na prvek nějakého většího celku, na prvek, o kterém je známo jeho chování vyjádřené v tomto případě chováním struktury jeho prvků z množiny ***E*** propojených prostřednictvím svých bran ***P*** spoji z množiny ***C***, tedy na prvek se strukturou ***(E, P, C)*** a branami ***(PI, PO)***. Stejně tak každý prvek ***E*** obvodu ***UUA*** může být pro další analýzu dekomponován na svou strukturu a na své (při pohledu na strukturu prvku primární) brány.

Definice 3.2:
Množinu obvodových prvků ***E*** lze rozdělit do těchto podmnožin:

E = (R ∪ MX ∪ FU), kde:
R je množina registrů,
FU je množina funkčních jednotek a
MX je množina multiplexorů**.**

V analýzy obvodu je důležité oddělit prvky se sekvenčním charakterem (registry) od prvků s charakterem kombinačním (funkční jednotky a multiplexory). Multiplexory mají mezi

kombinačními prvky význačnou roli spočívající v tom, že jednotlivé prvky propojují a zajišťují přenos dat mezi nimi.

Věta 3.1:
Podmnožiny ***R, FU*** a ***MX*** tvoří rozklad množiny ***E*** podle relace ekvivalence „být prvkem stejného druhu“, jsou tedy navzájem disjunktní a jejich sjednocení dává právě množinu ***E***.

Rozdělení množiny obvodových prvků do těchto tří tříd má význam pro analýzu testovatelnosti a odráží se v něm principy strukturovaného návrhu. Zatímco v množině registrů ***R*** jsou sdruženy prvky mající charakter paměťových prvků, které pak celému obvodu dávají sekvenční charakter (čistě kombinační obvod pak bude mít množinu ***R*** prázdnou), množiny ***FU*** a ***MX*** pokrývají zbytek obvodu, tj. kombinační sítě. Prvky ***MX*** slouží spíše při toku dat obvodem (přesně v duchu pravidel syntézy obvodů na úrovni RT s využitím multiplexovaných datových cest), prvky z množiny ***FU*** data zpravidla nějakým způsobem transformují, provádějí s nimi vhodné operace, jsou nositeli funkce celého obvodu. Tím ale není vyloučeno jejich využití k transportu diagnostických dat.

Definice 3.3:
P je množina bran prvků, kterou lze rozdělit na tři podmnožiny:

$P = (IN \cup OUT \cup CI)$, kde:
IN je množina vstupních datových bran,
OUT je množina výstupních datových bran a
CI je množina řídicích a synchronizačních vstupů.

Pro jednoduchost se předpokládá, že všechny brány z množin ***IN*** a ***OUT*** jsou vícebitové, stejné šířky (***m*** bitové), zatímco brány z množiny ***CI*** jsou typicky široké pouze 1 bit.

Množina ***P*** je, jak říká definice 3.1, množinou všech bran všech prvků v obvodě. Pro popis struktury obvodu je vlastně brána daleko významnějším pojmem než prvek. Tyto dva pojmy jsou spolu sice úzce svázány, ale je-li vzata do důsledku struktura obvodu, je to právě propojení bran spoji, které vytváří strukturu obvodu. Hovořit o propojení prvků je sice také možné, ale pro účely analýzy struktury jde o příliš hrubý pohled. Vždyť každý prvek má několik bran, často ne jen jednu ale více vstupních a výstupních.

Definice 3.4:
Nechť existuje funkce:

$\psi : E \rightarrow 2^P$,

která přiřazuje množinu bran obvodovému prvku:
(1) $\psi(e) = \{p \mid p$ je brána prvku $e\}$.
(2) Funkce ψ je definována pro všechny prvky z množiny ***E***.
(3) Musí platit, že $e_1 \neq e_2 \Leftrightarrow \psi(e_1) \cap \psi(e_2) = \varnothing$.
□

Funkce ψ tvoří vazbu mezi množinou prvků a množinou bran. Tato funkce určuje, která brána z množiny bran přísluší kterému prvku. Z fyzikálního významu funkce ψ plyne nutnost podmínky (2) a podmínky (3). Podmínka (2) zajišťuje, že v obvodě se nevyskytne prvek, u

kterého by nebylo možné říci, zda má nějakou bránu a jakou, podmínka (3) říká, že každý prvek z množiny P je funkcí ψ vázán k jedinému (obvodovému) prvku $e \in E$.

Věta 3.2:
Množina P je funkcí ψ rozdělena na $|E|$ disjunktních tříd.

Definice 3.5:
C je množina spojů:

$$C \subset (PI \cup P) \times (PO \cup P).$$

Jde vlastně o binární relaci v množině bran P a též mezi primárními vstupy a branami prvků a branami prvků a primárními výstupy, obsahuje uspořádané dvojice bran, mezi kterými existuje v obvodě spoj .

Věta 3.3:
Relace C je (což plyne z jejího fyzikálního významu):

- reflexivní,
- symetrická,
- transitivní.

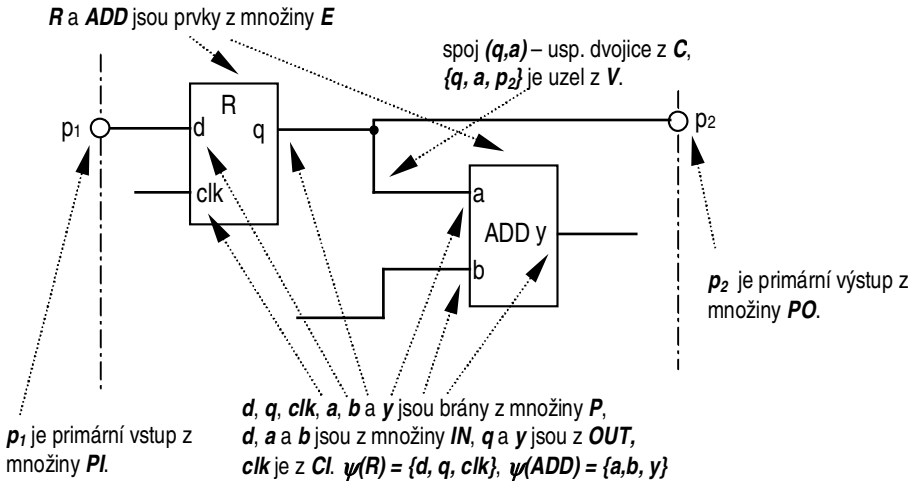
Je tedy vidět, že relace C je relací ekvivalence. Relace C indukuje rozklad na ekvivalenční třídy.

Prvek množiny C reprezentuje tedy formální model galvanického obvodového spoje několika bodů v obvodě. Při analýze struktury modelovaného obvodu jde přitom zpravidla o vyšetření, zda existuje spoj mezi dvěma body v obvodě. Proto byly zvolena forma binární relace, která může jednoznačně dát na takovou otázku odpověď. Fyzicky ovšem může nastat situace (a také v běžných obvodech často nastává), kdy jsou mezi sebou navzájem propojeny více než dva body v obvodě. Pak je logické, že všechny tyto body jsou spolu (vždy po dvojicích) navzájem v relaci. Lze též říci, že v jediném okamžiku mají všechny tyto body (a též spoj) stejnou hodnotu. Množina bodů obvodu, které jsou navzájem galvanicky propojeny spojem (jsou v relaci C) se bude dále nazývat *uzel*.

Definice 3.6:
Množina $V \subset (P \cup PI \cup PO)$ se nazývá *uzel*, jestliže $\forall p_1, p_2 \in V : (p_1, p_2) \in C$.

Věta 3.4:
Jestliže vstup a prvku e_1 je přímo (po *i cestě* délky 0) řiditelný z výstupu y prvku e_2, pak existuje jediný uzel, který je spojuje.

Definice 3.7:
Množina $D = \{0, 1, \uparrow, \downarrow\} \cup \{0,1\}^m$ je množina hodnot, které se na bráně mohou vyskytovat. Symboly $\uparrow$ a $\downarrow$ mají význam nástupné resp. sestupné hrany. Číslo m udává šířku datových cest obvodu, $\{0,1\}^m$ je tedy množina všech m-bitových binárních slov.



Obr. 3.1 Příklad k osvětlení formálního modelu obvodu.

Definice 3.8:
Nechť existuje zobrazení $\nu: P \rightarrow D$, kdy zápis $\nu(p) = w$ znamená, že na bráně $p \in P$ se vyskytuje hodnota $w \in D$

Všechny definice a věty zde uvedené jsou dokázány v literatuře [Ruz02]. Celý zde uvedený popis je zaměřen na syntézu obvodu s využitím strategie multiplexovaných datových cest. Právě rysy takto syntetizovaného obvodu reflektuje dále popsaný formální model obvodu. Model celého obvodu je tvořen uspořádanou pětici množin. Lze říci, že model má charakter grafu. Model umožňuje dále hierarchický popis, tato vlastnost však není v této práci využita. Je provedená základní klasifikace obvodových prvků s ohledem na jejich chování a roli v obvodě a též s ohledem na skutečnosti, které plynou z použití strategie syntézy s využitím multiplexovaných datových cest. Jedna z množin základní pětice modelující obvod je vlastně relace. Je to relace spojů mezi branami obvodových prvků. Vlastnosti této relace, diskutované v této kapitole, mají zásadní význam v další analýze testovatelnosti.

Využití struktury obvodu k přenosu diagnostické informace

Obvod na úrovni RT lze chápat jako struktura, ve které jsou jednotlivé obvodové prvky navzájem propojeny spoji. Prvky mohou zpracovávat data libovolné šířky (i celé slovo), za jediný spoj pak lze považovat i několik paralelně vedoucích vodičů v šířce slova. V obvodě lze jasně rozlišit datovou část, která realizuje vlastní funkci obvodu. Do ní data vstupují po datových primárních vstupech a výsledky činnosti obvodu lze najít na datových primárních výstupech. Druhou částí obvodu je řídicí část, řadič, který generuje řídicí signály pro prvky datové části v takovém časovém sledu, aby obvod vykonával svoji funkci a data v něm tekla správným směrem. Při testu datové části obvodu je původní (originální) obvodový řadič ve své funkci nahrazen řadičem testu.

Aplikace testu na obvod spočívá v přivedení testovacího vektoru na vstupy testované komponenty (části obvodu), získání odezvy komponenty na takový stimul a její vyhodnocení. Problém, jak přivést testovací vektor až do bodu, kam patří (často je hluboko ve struktuře

obvodu, běžně zvnějšku nedostupný) a vyvedení odezvy ven resp. do bodu, kam lze připojit analyzátor odezvy, vede často na dodatečné zásahy do struktury obvodu či její komplikace. Cílem této práce je nalézt a definovat jazyk, který je zadaný uspořádanou pěticí ***UUA***, jenž vyjadřující model struktury obvodu na úrovni meziregistrových přenosu (RT). Věty náležející do tohoto jazyka pak reprezentují ve struktuře obvodu takové části, které je možné využít v režimu testu obvodu k přenosu diagnostické informace (testovacího vektoru). Tímto způsobem je možné se vyhnout alespoň některým dodatečným úpravám obvodu (např. modifikace registru pro zařazení do řetězce *„scan"*), které mají smysl jen při testu a celý obvod prodražují, aniž by tyto úpravy vedly ke zkvalitnění funkce. V tomto směru se jeví jako velmi perspektivní využití stávajících datových cest v obvodě i pro přenos diagnostické informace. Obvod je přístupný ze svých primárních vstupů a výstupů. Skoro vždy vede nějaká datová cesta z určitého vstupu na nějaký výstup. Zpravidla však tato cesta mívá větší hloubku než jeden prvek a proto není otázka přístupnosti vnitřních komponent obvodu zcela jednoznačná. Počet možných i cest mezi primárním vstupem/výstupem a testovanou komponentou bývá velmi mnoho, proto by mohlo být výhodné formálně definovat i cesty v analyzovaném obvodu pomoci gramatiky. Všechny možné i cesty mezi zcela libovolnými prvky v obvodu se pak definovat jediným jazykem.

I cesty

Obecně lze říci, že obvod sestává z komponent (obvodových prvků) navzájem propojených prostřednictvím bran spoji. Spoje lze (za předpokladu jejich bezchybné funkce) považovat za transparentní. Transparentností se myslí taková schopnost přenosu dat mezi dvěma body v obvodu, kdy data vložená na jeden z bodů se objeví na druhém bodě beze změny.

Definice 4.1:
Pokud je možné za určitých podmínek z bodu p_1 do bodu p_2, kde $p_1 \in (IN \cup OUT \cup PI) \wedge p_2 \in (IN \cup OUT \cup PO)$, obvodu ***UUA*** přenést data beze změny, tedy nastane *i režim* (viz. dále) činnosti prvku obvodu platí $\forall d \in \{0, 1\}^m : v(p_1) = d \wedge v(p_2) = d$, říkáme, že mezi body p_1 a p_2 existuje *i cesta***.**

Množina $I = \{(p_1, p_2) | p_1 \in (IN \cup OUT \cup PI) \wedge p_2 \in (IN \cup OUT \cup PO) \wedge \forall d \in \{0, 1\}^m : v(p_1) = d \wedge v(p_2) = d\}$ je množinou všech možných *i cest* v obvodě ***UUA***.

Pokud se tyto dva body nachází na jediném spoji (jsou součástí jednoho uzlu), není většinou s transparentností problém. Problém je ovšem s *i cestami*, jejichž počáteční a koncový bod neleží ve stejném uzlu. Pak se na *i cestě* jistě objeví i nějaké obvodové prvky. U těch transparentnost není z principu zajištěna, neboť se předpokládá, že data určitým způsobem transformují, provádějí s nimi potřebné operace. U některých prvků však lze nalézt režim činnosti, kdy je prvek pro data transparentní, data přivedená na jeho vstup se objeví na jeho výstupu v nezměněné podobě. Tento režim činnosti je dále nazýván *i režim* (i jako identita). V tomto režimu je pak prvek využíván jen jako *transportér* dat. Toho lze s výhodou využít při aplikaci testu na obvod, kdy by bylo jinak nutno prvek obejít například úpravou obvodu. Jak získat množinu ***I*** napovídá věta 4.2, která bude uvedena dále v této kapitole.

Pro zpřehlednění dalšího zápisu se zavádí predikát vyjadřující, zda daná uspořádaná dvojice bran p_1, p_2 je propojena *i cestou* či nikoliv:

$(p_1,p_2) \in I \Leftrightarrow ipt(p_1,p_2)$.

Definice 4.2:
Pokud existuje režim činnosti prvku $e \in E$ takový, že v tomto režimu lze přenést data beze změny z některého jeho vstupu $in \in IN$ na některý jeho výstup $out \in OUT$, přičemž $in \in \psi(e) \wedge out \in \psi(e)$, tedy za určitých podmínek platí $\forall d \in \{0, 1\}^m: \nu(in) = d \wedge \nu(out) = d$, říkáme, že prvek e má *i režim* činnosti, uvnitř prvku existuje *i cesta* z jeho vstupu in na jeho výstup out.

Množina $M = \{(in, out) | in \in IN \wedge out \in OUT \wedge in \in \psi(e) \wedge out \in \psi(e) \wedge \forall d \in \{0, 1\}^m: \nu(in) = d \wedge \nu(out) = d\}$ je množinou všech vnitřních *i cest* v obvodě UUA.

Prvky, které mohou pracovat v *i režimu*, jsou pochopitelně součástí obvodu, do nějž jsou začleněny spoji. Lze pak najít řetězce transparentních prvků (zřetězených právě v obvodě již existujícími spoji) tvořících jakousi datovou cestu, po níž lze obvodem přenášet data při splnění určitých podmínek. Taková cesta bude též dále nazývána *i cestou*, neboť rovněž splňuje podmínky definice 4.1. Její délka, vyjádřená celým kladným číslem, má charakter metriky a odpovídá počtu transparentních prvků, přes něž vede. Za cenu mírné komplikace řadiče testu lze při vhodném využití *i cest* ušetřit mnohé jinak nevyhnutelné modifikace obvodu.

Věta 4.1:
Zřejmě $M \subseteq I$.

Definice 4.3:
Nechť existuje množina $FUI \subseteq FU$ taková, že $FUI = \{fui | fui \in FU \wedge \exists in, out: in \in \psi(fui) \wedge out \in \psi(fui) \wedge (in, out) \in M\}$.

Pro účely analýzy testovatelnosti je nutno rozlišovat mezi transparentními a netransparentními prvky. Proto je zavedena množina $FUI \subset FU$ funkčních jednotek, u kterých je nějakým způsobem možné nastavit *i režim*. U registrů (R) a multiplexorů (MX) se existence *i režimu* předpokládá.

Je pochopitelné, že ke každému *i režimu* se váže podmínka zahrnující jak vstupy (řídicí nebo datové), na nichž je *i režim* závislý, ale též hodnoty, které na tyto vstupy musí být přiloženy. Tato skutečnost je vyjádřena následujícím zobrazením:

Definice 4.4:
Zobrazení $\mu: M \rightarrow 2^{P \times D}$ je takové zobrazení, které každé *i cestě* uvnitř prvku (v *i režimu* prvku) přiřadí podmínky, za kterých tato *i cesta* (*i režim*) existuje.

Pro obvodový prvek $e \in E$ s datovým vstupem $i \in \psi(e) \wedge i \in IN$ a datovým výstupem $o \in \psi(e) \wedge o \in OUT$ (obě brány jsou široké m bitů) a je-li též $(i, o) \in M$, pak je zobrazení μ definováno:

$\mu(i, o) = \{(cp, x) | cp \in \psi(e) \wedge x \in D \wedge cp \neq i \wedge cp \neq o\}$, přičemž

$\forall d \in \{0,1\}^m$ platí, že když $\nu(i) = d \wedge ((\forall k \in \{1, ..., |\mu(i, o)|\}: (cp_k, x_k) \in \mu(i, o) \wedge \nu(cp_k) = x_k)$, pak také $\nu(o) = d$. Pak lze říci, že existuje *i režim* prvku e z jeho vstupu i na jeho výstup o za podmínek $\mu(i, o)$.

Relace μ přiřazuje každé *i cestě* uvnitř obvodového prvku (v *i režimu* obvodového prvku) množinu uspořádaných dvojic. Vždy první člen dvojice udává bránu podmiňující *i režim* a druhý člen hodnotu, která má na bráně být přítomna. Aplikace hodnot na brány tak, jak je udána v rámci množiny, aby měla účinnost, musí být provedena tak, aby se hodnoty na branách vyskytovaly současně, v jednom okamžiku.

Nastavení *i režimu*

Problém využití transparentnosti prvků je v nastavení *i režimu*. Nastavení *i režimu* není vždy jen otázkou nastavení nějakého řídicího vstupu na příslušnou úroveň. Jistě by se pro zjednodušení problematiky dal za *i režim* prvku pokládat pouze stav, kdy je prvek vlivem nějakého řídicího signálu „průchozí" právě po dobu trvání tohoto řídicího signálu. Takových prvků ale je vůči celkovému počtu prvků relativně málo a pokud už se v obvodě vyskytují, nebývají příliš zřetězeny. Z toho důvodu by při využití pouze takových prvků bylo možno nalézt pouze nedostatečné množství jen krátkých a tudíž obtížně využitelných *i cest*.

Přitom při zachování určitých pravidel lze pojem *i režimu* zobecnit a využít tak k přenosu diagnostické informace některých dalších vlastností obvodových prvků. Například je možné beze změny přenést data přes registr. Vždyť registry z principu ani data měnit nemají. Jenže u registru běžně není režim, kdy by se data vložená na jeho vstup přímo objevovala na jeho výstupu. Výstupy odráží zpravidla stav vnitřních klopných obvodů, ve kterých je informace uložena až do okamžiku příchodu dalšího hodinového impulsu, který je přepíše aktuálním stavem vstupů. Na výstupu registru jsou tedy data ze vstupu z okamžiku příchodu hodinového impulsu. To silně připomíná transparentní chování prvku až na to, že okamžik „transparentnosti" je velmi krátký. Protože však je tento okamžik dobře ovlivnitelný, lze podobné chování též považovat za *i režim*.

Rovněž u prvků provádějících aritmetické a logické operace je možné nalézt režim činnosti, který lze považovat za transparentní.

Z hlediska nastavení *i režimu* lze potenciálně transparentní prvky rozdělit takto:

- Prvky s *i režimem* závislým na hladinových řídicích vstupech,
- prvky s *i režimem* závislým na hranových řídicích (resp. synchronizačních) vstupech a
- prvky s *i režimem* závislým na datových vstupech.

Věta 4.2:
Nechť $M \subset (PI \cup PO \cup P)$ **x** $(PI \cup PO \cup P)$ a nechť $C \subset (PI \cup PO \cup P)$ **x** $(PI \cup PO \cup P)$, pak množina všech dvojic bran v obvodě, mezi kterými existuje *i cesta*, je $I=\{(p_1, p_2) | (p_1, p_2) \in J \wedge J=(C \cup M)^* \wedge p_1 \in (IN \cup OUT \cup PI) \wedge p_2 \in (IN \cup OUT \cup PO)\}$.

Struktura cest v obvodě pro účely analýzy testovatelnosti může být vyjádřena grafem $G=((PI \cup PO \cup IN \cup OUT), (C \cup M))$, přičemž množina $(PI \cup PO \cup IN \cup OUT)$ je množinou vrcholů a množina $(C \cup M)$ je množinou hran tohoto grafu. *I cesty* jsou pak skutečně cesty (ve smyslu teorie grafů) právě v tomto grafu G. Pokud však množinu $(PI \cup PO \cup IN \cup OUT)$ budeme považovat za množinu terminálních symbolu a množinu $(C \cup M)$ použijeme jako předlohou pro tvorbu přepisovacích pravidel, tak jsme schopni pomocí algoritmu 1 definovat formální model konstrukce gramatiky G jazyka *i cest*. Orientovaný graf G může obsahovat

cyklus, ten však v obvodu fyzikální podstatu, proto je formální model konstrukce zvolen tak, aby jej zamezil.

Algoritmus 1:
Konstrukce gramatiky G = (N, T, R, S). Jazyk L(G) generovaný gramatikou, je jazykem i cest, mezi vstupní branou InI a výstupní branou OutI prvku obvodu.

T konečná abeceda, **T = {PI ∪ PO ∪ IN ∪ OUT},**
N konečná množina neterminálních symbolů
S počáteční symbol gramatiky, **s ∈ N**
R konečná množina přepisovacích pravidel
InI brána, ze které se má vést *i cesta*, **InI ∈ T**
OutI brána, kde končí *i cesta*, **OutI ∈ T**

Vstup: **T** je konečná množina všech bran v obvodě (datových), mezi kterými může existovat *i cesta*, **T = {PI ∪ PO ∪ IN ∪ OUT }**
Rel množina všech dvojic bran v obvodě, mezi kterými existuje i cesta, Rel = C ∪ M
InI vstupní brána, **InI ∈ T**
OutI výstupní brána, **Out ∈ T**

Výstup: **N** množina nonterminálních symbolů
R množina přepisovacích pravidel tvaru $A \rightarrow xB$, $A,B \in N, x \in T$

Metoda:

1. množina nonterminálů $N = \{S\} \cup \{ <A,X> \mid A \in T, X \subseteq T \}$
2. $R_1 = \{ S \rightarrow a < \{a\} , a > \mid < \{a\} , a > \in N \wedge a \in InI \}$
3. $R_2 = \{< X , b > \rightarrow \varepsilon \mid X \subseteq T \wedge b \in OutI \}$
4. $R_3 = \{< X , a > \rightarrow b < X \cup b , b > \mid X \subseteq P \wedge a \in T \wedge b \in T \wedge (a , b) \in Rel \wedge$ 1. $b \notin X \}$
5. množina přepisovacích pravidel $R = R_1 \cup R_2 \cup R_3$

Pomocí algoritmu 1 lze sestrojit k libovolně zadaným branám gramatiku G, jenž bude generovat jazyk L (G) všech možných *i cest* v analyzovaném obvodu. Konstrukce algoritmu by byla mnohem jednodušší pokud bychom v gramatice povolovali cyklus či rekurzi. Pak by jazyk L(G) byl nekonečný. Věty jazyka (*I cesty*), které byly tvořený rekurzí nemají v obvodě fyzikální ani logickou podstatu, proto jsou z jazyka všech *i cest* záměrně vyloučeny.

Vlastnosti gramatiky G všech *i cest* v *UAA* obvodu sestrojené pomocí algoritmu 1:

- Gramatika neobsahuje cyklus ani rekurzi. Plyne z konstrukce přepisovacích pravidel a neterminálních symbolů. Tudíž jazyk L(G) generovaný gramatikou je konečný. Lze dokázat že pro každý konečný jazyk lze sestrojit regulární gramatiku.
- Množina přepisovacích pravidel R gramatiky G obsahuje pravidla tvaru $A \rightarrow xB$, $A,B \in N, x \in T$. Jedná se tudíž o regulární gramatiku.
- Formální model konstrukce gramatiky G vytváří mnoho zbytečných neterminálních symbolů, pravidel a též i zavádí zbytečná e-pravidla. Pokud bychom požadovali vlastní gramatiku, tak aplikujeme na získanou gramatiku G známé algoritmy pro

odstranění zbytečných pravidel, symbolu a též i algoritmus pro odstranění *e-pravidel* (e-pravidlo ($S \rightarrow \varepsilon, S \in N, \varepsilon \in T$) může být aplikováno pouze ve startovacím symbolu S gramatiky G). Navíc tato gramatika bude splňovat podmínku pravě lineárních gramatik.

Uvedený způsob konstrukce všech *i cest* pomocí gramatik ukazuje jak lze sestrojit množinu přepisovacích pravidel gramatiky, z libovolné vstupní brány **InI** na libovolnou výstupní bránu **OutI**. Pro definici jazyka všech *i cest* s libovolnou vstupní a výstupní branou stanovíme **InI = T** a **OutI = T**. Tento jazyk, je jazykem všech možných *i cest* v analyzovaném obvodě *UAA*.

Řetězce jazyka všech *i cest*, pak lze analyzovat z různých hledisek. Jako jedno z důležitých kritérií lze považovat délku řetězce, jenž reprezentuje počet bran v obvodu, jimiž prochází konkrétní *i cesta*. Za další kriterium pro klasifikaci řetězců jazyka by se dalo považovat výskyt či absence některých terminálních symbolů. Například, může být požadováno vzhledem ke složitosti řadiče testu, aby diagnostika obvodu byla provedena s minimálním počtem prvků s *i režimem* závislým na datových vstupech. K těmto prvkům musí vézt *i cesty*, pomocí nichž lze ovládat řídící vstupy pro zajištění *i režimu* prvků. Pokud požadujeme provést diagnostiku s nízkou spotřebou, vybereme *i cesty* takové, které v celkovém součtu spotřeby jsou nejméně náročné. Jiné požadavky na *i cesty* může uplatňovat diagnostika, kterou chceme uplatnit co nejrychleji. Tento druh testu obvodu především požaduje, aby *i cesty* procházely různými branami. Pak lze provést test několika prvků současně.

Shrnutí

V prací je prezentován formální přístup k problému popisu analyzovaného obvodu, který byl převzat z literatury [Ruz02]. Pojem *i cesty* a veškeré pomocné pojmy jsou zde formálně definovány, též je v prácí uvedena úloha prvků na *i cestách*, které musejí pracovat v tzv. *i režimu* a rovněž je i zjednodušeně uvedeno jak tento režim nastavit. Díky jednoduchému formálnímu modelu postavenému na principech teorie grafů je relativně snadná i úloha hledání *i cest* v obvodě. Hlavně je v této prácí uveden nový pohled na problematiku popisu a definice *i cest* v analyzovaném obvodě. Díky tomu lze problém popisu *i cest* mezi různými body obvodu (branami) převést i na problematiku teorie jazyků. Věty jazyka se pak dají jednoduše analyzovat a třídit podle různých pohledů, které jsou z hlediska diagnostiky obvodu podstatné (délky řetězce, výskyt či absence určitých terminálních symbolů – prvků obvodu).

Literatura

[Ruz02] Růžička, R.: Formální přístup k analýze testovatelnosti číslicového obvodu na úrovní RT [disertační práce]. FIT VUT v Brně, 2002, 102 stran

[Kot99] Kotásek, Z.: Uplatnění principů řiditelnosti/pozorovatelnosti při návrhu číslicových obvodů [habilitační práce]. FEI VUT v Brně, 1999, 80 stran