

Úvod do diagnostiky číslicových systémů

studijní text frekventantů předmětu PV171/3

Obsah

1 Diagnostika číslicových systémů.....	3
1.1 Poruchy číslicových systémů	4
1.1.1 Klasifikace poruch číslicových systémů	4
1.1.2 Modely poruch číslicových systémů	8
1.2 Testování číslicových systémů.....	9
1.3 Sestavení kroku testu	11
1.3.1 Postup sestavení kroku testu.....	11
1.3.1.1 Analýza zapojení testovaného obvodu.....	12
1.3.1.2 Tabulková metoda.....	12
1.3.1.3 Metoda citlivé cesty.....	15
1.3.1.4 Metoda Boolovské diference.....	20
1.3.1.5 Metoda D-algoritmu.....	25
1.3.2 Doslov k sestavení kroku testu	46
1.4 Sestavení testů kombinačního obvodu.....	47
1.4.1 Sestavení detekčního testu.....	47
1.4.2 Sestavení lokalizačního testu.....	50
1.5 Příznaková analýza	55
1.5.1 Základ příznakové analýzy	56
1.5.2 Generování příznaku	58

1 Diagnostika číslicových systémů

Diagnostika obecně představuje činnosti směřující ke zjištění stavu zkoumaného objektu. Tato činnost je aplikována v řadě oborů lidské činnosti a ze zcela prozaických důvodů má velmi dlouhou tradici. Vznik prvních diagnostických metod lze vysledovat již v počátcích lidské civilizace, kdy vznikaly první větší státní útvary a kdy lidé začali organizovaně používat poznatky o přírodě a nejrůznější technické systémy ve svůj prospěch. V této souvislosti stačí připomenout lékařství. Naším zájmem bude diagnostika technických systémů se zaměřením na metody související se zjišťováním stavu číslicových systémů.

Diagnostika, tedy testování a ověřování správné činnosti číslicových systémů má oproti klasickým systémům složených z lineárních obvodů zvláštnosti vyplývající z principu činnosti základních komponent těchto systémů, tedy ze specifických principů činnosti číslicových obvodů. Je třeba si uvědomit, že se jedná o prověřování správné činnosti technického vybavení číslicového systému. Prověřování programového vybavení číslicových systémů představuje odlišnou úlohu.

Diagnostika číslicových systémů je založena na principu determinismu funkcí, poruch a struktury systému jako celku. Jakákoliv porucha konstrukčního prvku systému je tedy detekovatelná, projeví se odchylkou funkce systému jako celku.

Systém lze rozčlenit na jednotlivé konstrukční bloky, které pokud pracují, nemohou způsobit chybnou funkci systému jako celku. Základním předpokladem diagnostického postupu je tedy defragmentace systému na dílčí podsystémy a posléze prověřování stavu jednotlivých podsystémů. Hloubka defragmentace se může blížit až na úroveň konstrukčních prvků číslicového systému.

Číslicové obvody pracují s fyzikálními veličinami interpretujícími logické hodnoty v diskrétním tvaru, nemají tudíž spojitý charakter v čase. Číslicové obvody jsou konstruovány tak, aby na svém výstupu dokázaly regenerovat ne zcela standardní průběhy vstupních signálů. Z principu jejich činnosti vyplývá, že na hodnotě výstupní proměnné se jednotlivé části obvodu podílejí v závislosti na vstupních proměnných. Jinak řečeno jednotlivé části obvodu se na výstupní funkci podílejí pouze v určitých časových okamžicích.

Číslicové obvody se podle své struktury dělí na sekvenční číslicové obvody a na kombinační číslicové obvody. Ze struktury sekvenčních obvodů je zřejmé, že diagnostika obecných sekvenčních obvodů je podstatně komplikovanější než diagnostika kombinačních obvodů. V praxi je proto snahou komplikace vyplývající z nepřijatelných vlastností

sekvenčních obvodů eliminovat vhodným konstrukčním řešením těchto částí číslicového systému.

Prověřování správné činnosti číslicového systému se provádí testy - sekvencemi dat - které jsou sestaveny tak, aby pokud možno byly prověřeny všechny dílčí funkce technického vybavení číslicového systému. Test číslicového systému je tvořen sekvencí kroků testu - sekvencí vstupních údajů a vybraných funkcí testovaného systému, které jsou systémem zpracovávány a prováděny. Pokud systém nepracuje správně, poskytuje odlišné výsledky ve srovnání se správně pracujícím systémem. Základním předpokladem diagnostiky číslicových systémů je předpoklad, že porucha každého konstrukčního prvku je detekovatelná.

Základní úlohou diagnostiky číslicových systémů je najít a sestavit test - posloupnosti kroků testů - zveřejňující nepochybným způsobem poruchu číslicového systému. Nalezení a odstranění poruchy číslicového systému je až následujícím krokem, i když ani tento krok není v praxi zcela triviální.

1.1 Poruchy číslicových systémů

Poruchy, vyskytující se v číslicových systémech mají různé fyzikální nebo chemické příčiny. Pro potřeby diagnostiky logických obvodů se oprostíme od fyzikálních a chemických procesů způsobujících chybné chování číslicových obvodů a budeme se zabývat pouze jevovou stránkou chování porušených obvodů. Nebudeme se tedy zabývat jejich podstatou, ale jejich projevy.

1.1.1 Klasifikace poruch číslicových systémů

Klasifikace poruch číslicových obvodů může být velice široká. Většina kritérií vychází z jejich vlivu na fungování obvodu, případně z časových charakteristik jejich projevů. Klasifikací poruch se zabývají také technické normy. Při popisu funkce elektronického zařízení se ve smyslu názvoslovné normy používají termíny **porucha** (ang. fault) a **chyba** (ang. error).

Porucha je jev spočívající v ukončení schopnosti objektu plnit požadovanou funkci podle zadaných technických podmínek.

Chyba je pak rozdíl mezi správnou a skutečnou hodnotou libovolné veličiny, která své hodnoty nabývá díky funkci objektu.

Chyba je tedy důsledkem vzniku poruchy, projevem poruchy. Zároveň je zřejmé, že ne všechny poruchy se musí nezbytně projevit. Zda se porucha projeví, závisí na tom, jestli se konstrukční prvek postižený poruchou bude podílet na realizaci řešení konkrétní úlohy. Je tedy zřejmé, že vznik poruchy konstrukčního prvku se nemusí ihned projevit chybou,

jinak řečeno, doba vzniku poruchy nemusí souhlasit s dobou projevu chyby. Porucha konstrukčního prvku zařízení se projeví až v okamžiku, kdy se tento prvek má podílet na realizaci některé dílčí funkce elektronického zařízení.

Rozlišení poruchového a bezporuchového stavu elektronického zařízení se provádí úkony, které se označují termínem **testování**. Zjišťování stavu elektronického systému bývá zpravidla rozděleno do dvou fází. V první fázi je základním úkolem zjistit, zda porucha elektronického zařízení nastala či nikoliv, ve druhé fázi je hlavním problémem nalezení (lokalizace) místa poruchy. Obě fáze zjišťování stavu elektronického systému analyzují projevy již existující poruchy a nezabývají se problémem předcházení vzniku poruchy elektronického zařízení. Tímto problémem se zabývá **predikční diagnostika**.

Predikční diagnostika elektronických systémů si na rozdíl od klasické diagnostiky klade za cíl určovat pravděpodobnost vzniku poruchy elektronického systému v definovaném čase, přičemž je brán do úvahy jeho aktuální technický stav.

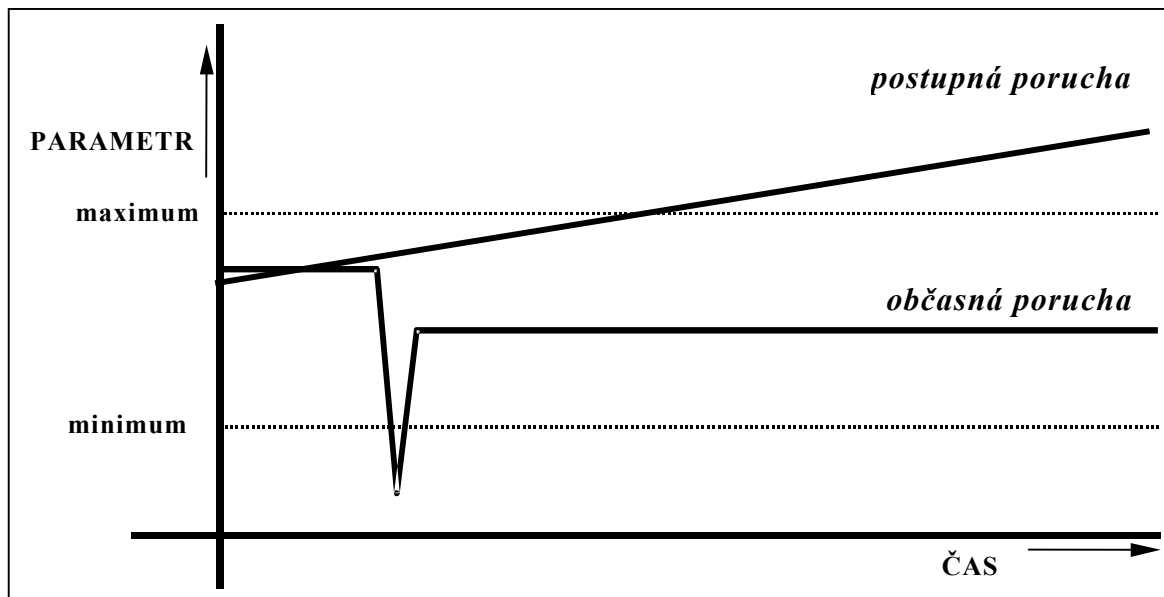
Technický stav elektronického zařízení se odráží ve výskytu poruch nejrůznějšího charakteru. Typologií poruch se zabývá norma a rozlišuje následující poruchy:

- náhlá porucha,
- postupná porucha,
- občasná porucha,
- úplná porucha,
- částečná porucha,
- havarijní porucha (náhlá a úplná),
- degrační porucha (postupná a částečná).

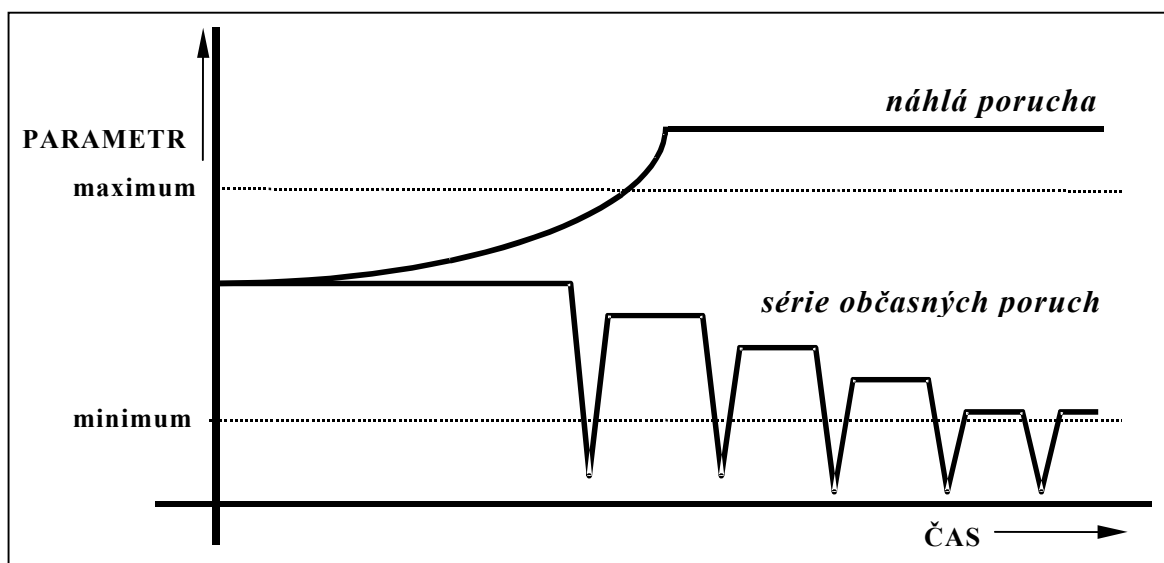
Charakteristické vlastnosti některých typů poruch jsou schematicky zobrazeny na obr 1.1 a 1.2.

V zařízení bez poruchy nabývají parametry signálů přenášejících informaci hodnot, které se pohybují uvnitř oblastí omezených mezními, povolenými hodnotami. U číslicových systémů platí, že parametry signálů se uvnitř vymezených oblastí nenacházejí nepřetržitě, ale pouze v určitých, přesně definovaných okamžicích. Pro číslicové obvody jsou definovány statické a dynamické parametry. Na změny obou typů parametrů mají vliv fyzikální a chemické procesy probíhající v materiálech z nichž jsou vytvořeny konstrukční prvky elektronického zařízení. Tyto procesy způsobují, že pracovní body konstrukčních prvků se přibližují k mezním hodnotám a tím se k mezním hodnotám posouvají parametry signálů přenášejících informaci. Takto se konstrukční prvky stávají citlivější na další fyzikální a chemické jevy, výsledkem čehož je

dočasný a později trvalý posun některého parametru signálu nesoucího informaci mimo povolenou oblast - tedy vznik poruchy.



Obr. 1.1: Základní typy poruch číslicových systémů



Obr. 1.2: Základní typy poruch číslicových systémů

Vysoký počet poruch v elektronických systémech má dočasný charakter, jedná se o občasně poruchy, přičemž se četnost těchto poruch mění. Příčina těchto poruch spočívá ve fyzikálních jevech souvisejících se šířením elektrických impulsů po vedení a na fyzikálních vlastnostech vodičů sloužících k šíření těchto impulsů mezi jednotlivými obvody elektronického zařízení. Každý vodič elektrických signálů parazitně vyzařuje elektromagnetické záření do okolního prostoru, ale také přijímá elektromagnetické záření z okolního prostředí. Přeslechy a odrazy na

nepřízpůsobených vedeních se mohou navzájem sečítat a odečítat s užitečným signálem a takto ovlivňovat přenášenou informaci.

Za účelem zvýšení účinnosti a efektivity diagnostiky elektronických systémů je velmi zajímavé zabývat se prognostickými metodami zjišťování stavu systémů - predikční diagnostikou jejich poruch. Včasně provedenou opravou konstrukčního bloku systému lze předejít poruše, která sama o sobě nemusí znamenat katastrofu pro funkci systému jako celku, ale která způsobí prostoj systému v takové délce a v takovém okamžiku, během něhož mohou nastat katastrofické následky pro okolní prostředí v jehož prospěch systém pracuje. Právě predikční diagnostika si klade za cíl vzniku těchto následků předcházet.

Predikční diagnostika elektronických systémů řeší dvě základní úlohy. První úloha představuje vyhledávání signifikantních a pokud možno především deterministických parametrů - **příznaků - markerů**, které lze schůdným způsobem měřit a zaznamenávat a které umožňují s dostatečnou přesností a s definovatelnou pravděpodobností predikovat chování sledovaného systému a tak upozornit na možnost vzniku jeho poruchy. Druhou, hlavní úlohou predikční diagnostiky je na základě znalosti chování markerů vyčíslit pravděpodobnost vzniku poruchy v definovaném čase a v závislosti na momentálním technickém stavu elektronického zařízení. Predikční diagnostika tím, že respektuje aktuální, individuální stav testovaného zařízení, umožňuje poskytovat odpověď na řadu otázek spojených s provozem a údržbou elektronického zařízení.

Predikce vzniku poruchy a její včasná eliminace je užitečná u řady společensky a ekonomicky důležitých systémů. V případě informačních systémů lze predikcí vzniku poruchy omezit výpadky ve funkci informačního systému a tímto způsobem zvýšit dostupnost informací pro běžného uživatele. Obdobně tomu může být u různých dispečerských systémů, technologických celků, zbrojních systémů atd.

Pro zjednodušení diagnostické úlohy pracujeme s tzv. modely poruch. V číslicových systémech má pouze určitý počet poruch charakter poruchy trvalé, tedy poruchy nezávislé na rychlosti jejich synchronizace. V reálných zařízeních se s větší pravděpodobností vyskytují dočasné poruchy související s frekvenční závislostí některých elektrických parametrů konstrukčních prvků. Z praktického hlediska je tedy racionální testovat číslicové obvody především v dynamickém režimu při předepsané frekvenci hodinových pulsů systému v němž je použit. Rychlost přivádění souboru testovacích vektorů a snímání a vyhodnocování výstupních kombinací testovaného obvodu (diagnostických příznaků) je závislá na technickém vybavení testovacího pracoviště, které musí být vybaveno zařízením pro detekci dočasných stavů testovaného systému.

1.1.2 Modely poruch číslicových systémů

Nejběžnějšími modely poruch jsou porucha typu stálá logická **0** (označujeme ji symbolem t_0) a porucha typu stálá logická **1** (označujeme ji symbolem t_1).

Název těchto modelů je velmi výstižný - představují stav, kdy porucha v příslušném uzlu logického obvodu (tedy na vodiči spojujícím jeden výstup logického obvodu s více vstupy následující vrstvy logických obvodů) způsobuje takový stav, který následující logické prvky obvodu interpretují jako vstupní logickou **0** nebo logickou **1** a dále s ní podle své funkce pracují. Řadu konkrétních poruch lze takto interpretovat a diagnostickou úlohu zjednodušují. Poruchu chápeme jako změnu logické struktury testovaného obvodu - v tomto případě jako přerušení propojení mezi prvky obvodu a přivedení příslušné logické hodnoty na vstup prvku bezprostředně navazujícího na místo s poruchou.

Tyto modely dovolují popsat celou řadu reálných poruch a pro potřeby technické diagnostiky jsou postačující. Modely poruch t_0 a t_1 lze bez problémů použít pro modelování trvalých i dočasných poruch logických obvodů.

Je nutno si uvědomit, že poruchu typu t_0 nebo t_1 v uzlu logického obvodu může způsobit jak porucha výstupu zdroje logického signálu připojeného do uzlu, tak poruchy vstupů následující vrstvy logických obvodů. Nalezení konkrétního zdroje chybného chování - tedy poruchy - vyžaduje detailní měření elektrických parametrů v uzlu logického obvodu, což překračuje definované podmínky testování a tudíž je mimo možnosti dále popisovaných metod.

Komplikovanější je situace u poruch typu zkrat. I v tomto případě lze použít logiku, že porucha znamená změnu struktury logického obvodu. Interpretace zkratu je poněkud obtížnější. Pokud se jedná o zkrat signálového vodiče na zemnicí vodič nebo na napájecí napětí, je interpretace obdobná jako u modelů t_0 a t_1 . Zkrat mezi signálovými vodiči nemá jednoznačnou interpretaci. Vždy záleží na typu použitých logických prvků. Hlavním aspektem je výstupní impedance logického obvodu ve stavu **log.1** a **log.0**. Podmínky na zkratovaných vodičích s navzájem inverzními hodnotami signálu se budou blížit výstupnímu stavu s nižší výstupní impedancí. Může se stát, že hodnoty signálů na vodičích s poruchou typu zkrat mezi signálními vodiči nebudou odpovídat předepsaným technickým parametrům, ale rozhodující je, jak tyto hodnoty interpretují vstupy logických prvků bezprostředně připojených ke zkratovaným signálním vodičům. Popsanou situaci vyjadřují tabulky 1.1 a 1.2.

Z tabulek je zřejmé, že pokud je výstupní impedance nižší pro **log.0**

na výstupu, pak se zkrat mezi signálními vodiči chová jako logický součin (zkrat ve schéma obvodu nahradíme poruchovým obvodem **AND**). Pokud je výstupní impedance nižší pro **log.1** na výstupu, pak zkrat mezi signálními vodiči nahradíme logickým součtem (zkrat ve schéma obvodu nahradíme poruchovým obvodem **OR**).

Y1	Y2	zkrat
0	0	0
0	1	0
1	0	0
1	1	1

Tab. č.:1.1

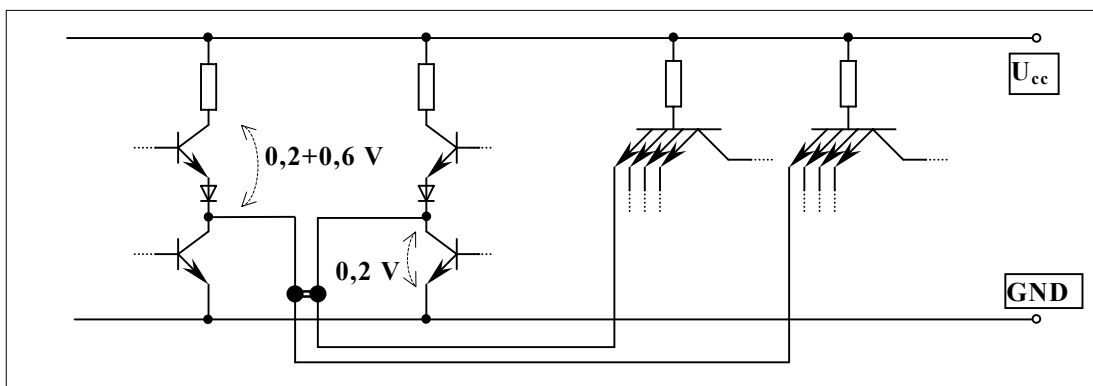
Nižší impedance pro log.0 na výstupu

Y1	Y2	zkrat
0	0	0
0	1	1
1	0	1
1	1	1

Tab.č.: 1.2

Nižší impedance pro log.1 na výstupu

Jako příklad je na obrázku 1.3 zobrazeno chování obvodů technologie TTL zkratovaných na výstupech. Výstup levého obvodu je nastaven na logickou "1" (je otevřen horní tranzistor výstupní části obvodu), zatímco výstup druhého obvodu je nastaven na logickou "0" (je otevřen dolní tranzistor výstupní části obvodu). Díky zkratu na vodičích připojených k výstupu se na obou výstupech ustálí napětí 0,2 V (tedy logická "0"). Zkrat na signálových vodičích obvodů typu TTL se chová jako **AND** (logický součin).



Obr. 1.3: Rozbor chování zkratu mezi signálními vodiči u obvodu typu TTL - Dvojitá čára představuje zkrat

1.2 Testování číslicových systémů

Kontrolu funkce číslicových systémů můžeme provádět ve *statickém*, *zrychleném* nebo v *dynamickém* režimu. Pro potřeby výkladu se budeme zabývat diagnostikou trvalých poruch. Pro hledání dočasných poruch se používají podobné metody jako pro hledání trvalých poruch, ale je třeba uvažovat skutečnost, že taková porucha se projeví nezávisle na začátku testování, jinak řečeno projeví se náhodně jednou nebo vícekrát v některé z n-realizací testu. Kvalitní testovací pracoviště musí být vybaveno měřicí technikou schopnou registrovat tato náhodná vybočení

ze správné činnosti obvodu. Vhodným prostředkem jsou paměťové osciloskopy nebo logické analyzátory.

Testy poruch číslicových obvodů lze dále provádět za různých stresujících provozních podmínek, např. při různé teplotě, napájecím napětí, frekvenci hodinových impulsů atd.

Testy číslicových systémů se skládají z posloupnosti kombinací vstupních signálů testovaného obvodu. Každá kombinace vstupních signálů představuje **krok testu - testovací vektor**. Jejich vhodným řazením lze dosáhnout specifických vlastností testů.

Podle způsobu sestavení testů můžeme rozlišovat tyto druhy testů:

test triviální - jedná se o test, kdy na vstup obvodu jsou postupně přivedeny všechny možné vstupní testovací vektory. U menších obvodů tato skutečnost nepřináší žádné komplikace. U rozsáhlejších obvodů však rozsah tohoto testu může nabýt značného objemu a je žádoucí provést minimalizaci počtu testovacích vektorů. Možnost minimalizace vyplývá ze skutečnosti, že pro indikaci některých poruch lze použít více vektorů a naopak, že jeden vektor indikuje více poruch,

test strukturní - vychází ze zapojení testovaného obvodu, tedy sestavení testovacího vektoru není prováděno na minimalizované logické rovnici funkce testovaného obvodu. Je možno sestavit test pro zadanou množinu poruch, která odpovídá skutečným poruchám vyskytujícím se v reálném obvodu. K sestavení strukturního testu je však potřeba znát vnitřní strukturu obvodu až na úroveň hradel a dalších konstrukčních prvků. Z tohoto důvodu lze ze strukturního testu sestavit lokalizační test,

test funkční - vychází z popisu funkce obvodu. Pro sestavení funkčního testu se používá minimalizovaná logická rovnice obvodu. Funkční test po minimalizaci představuje detekční test obvodu,

test detekční - posloupnost testovacích vektorů, které na výstupu testovaného obvodu vygenerují posloupnost signálů jednoznačně indikující správnou nebo chybnou funkci obvodu. Jedná se o minimalizovanou verzi triviálního, strukturního nebo funkčního testu,

test lokalizační - podmíněná posloupnost testovacích vektorů. Názorně se zobrazuje tzv. diagnostickým stromem. Sestavuje se ze strukturního testu obvodu. Umožňuje pouze na základě posloupnosti logických úrovní výstupů testovaného obvodu lokalizovat všechny možné nebo pravděpodobné poruchy obvodu. Vzhledem k překrývání projevů jednotlivých poruch nelze dosáhnout 100% rozlišitelnosti lokalizace poruch,

testy příznakové analýzy - umožňují rychlé a snadné otestování obvodu, vyžadují však určité předchozí konstrukční úpravy. Testy využívají programovatelný generátor posloupnosti a příznakový analyzátor. Využívají vhodné vlastnosti příznakové analýzy, přičemž provádějí jak

funkční, tak do jisté míry i strukturní testování obvodu. Princip činnosti příznakové analýzy bude popsán v kapitole 1.6. Testy příznakové analýzy se především používají pro rychlé testování obvodů pomocí jehlového pole, které se přitlačí na plošné spoje obvodové realizace s konkrétními obvody. Programovatelný generátor vstupních posloupností může pracovat v několika režimech:

- generování binárního kódu (nejjednodušší realizace),
- generování *Grayova kódu* (nejstabilnější výsledky - po sobě následující vzorky vstupních kombinací se liší pouze v jednom bitu),
- generování zvolených testů (mohou být problémově orientovány - určeny pro hledání pouze určité skupiny nejpravděpodobnějších poruch).

1.3 Sestavení kroku testu

Testování logických obvodů je činnost, která umožňuje stanovit technický stav logického obvodu, tedy zda obvod plní nebo neplní předepsanou funkci. Logické obvody se dělí na kombinační a na sekvenční obvody. Testování sekvenčních obvodů komplikují paměťové elementy začleněné do jejich struktury. Tématem této kapitoly je testování kombinačních obvodů, budeme tedy hovořit pouze o kombinačních obvodech, a proto již tento fakt nebudeme v textu dále zdůrazňovat.

Pro potřeby testování číslicových obvodů se uvažuje situace, kdy na testovaný obvod pohlížíme jako na uzavřený systém. Máme přístup pouze ke vstupním a výstupním kontaktům obvodu a komponenty vlastního obvodu a spoje mezi nimi (uzly) jsou nedostupné. Nemůžeme tedy sledovat odezvy vnitřních uzlů systému na vstupní signály testovaného obvodu a jsme tudíž odkázáni pouze na sledování odezev výstupu. Na testovaný obvod tedy nahlížíme jako na tzv. "Black Box" a řešíme úlohu, kdy podle odezvy výstupu na vstupní signály usuzujeme na vnitřní strukturu testovaného obvodu.

V případě testování číslicových obvodů je situace usnadněna znalostí zapojení bezchybného obvodu. Na základě tohoto zapojení lze odvodit výstupní hodnoty bezchybného obvodu, které budeme pro potřeby dalšího výkladu nazývat **etalon**. Etalonové hodnoty výstupu logického obvodu mohou být změněny poruchou obvodu. Příčiny poruch číslicových obvodů jsou velice pestré a nepatří do probírané tematiky.

1.3.1 Postup sestavení kroku testu

Je zřejmé, že libovolná porucha v libovolném místě obvodu může způsobit změnu chování obvodu. Tento jev platí bez výjimky pro

minimalizované logické obvody. U neminimalizovaných struktur se může stát, že některé poruchy mohou být zamaskovány vnitřními redundantními strukturami obvodu, případně mohou splývat s jinými poruchami (jejich projev na výstupu testovaného obvodu je totožný). Podle logiky předchozího konstatování lze očekávat, že porucha obvodu změni výstupní hodnotu obvodu alespoň pro jednu vstupní kombinaci. Takovou vstupní kombinaci budeme nazývat **krok testu**.

Krok testu je taková vstupní kombinace testovaného obvodu, která v uzlu s poruchou nastaví inverzní hodnotu signálu než způsobuje porucha a která umožní, aby tato změněná hodnota mohla být demonstrována na výstupu testovaného obvodu. Toto je princip sestavení kroku testu z něhož vycházejí jednotlivé metody sestavení kroku testu.

V dalších kapitolách bude popsáno pět základních metod sestavení kroku testu - analýza zapojení testovaného obvodu, tabulková metoda, metoda citlivé cesty, Boolovská diference a tzv. D-algoritmus.

1.3.1.1 Analýza zapojení testovaného obvodu

Tato metoda vyžaduje hluboké znalosti odborníka provádějícího testování číslicového obvodu. K nalezení kroku testu jsou potřeba podrobné popisy a schémata systému. Diagnostik stanovuje vstupní kombinace proměnných na základě svých zkušeností a intuice tak, aby na výstupu byla pozorovatelná změna chování výstupních signálů.

Tuto metodu zde uvádíme pouze pro úplnost textu a vzhledem k vysokým odborným nárokům na znalosti a schopnosti obsluhy se jí nebudeme dále zabývat.

1.3.1.2 Tabulková metoda

Tabulková metoda má za základ jednu z metod zobrazení logické funkce obvodu. Její princip spočívá v sestavení tabulky vyjadřující reakci obvodu bez poruchy a s poruchou na všechny kombinace vstupních proměnných obvodu s poruchou. Testovacím vektorem je ta vstupní kombinace, pro kterou se liší výstup testovaného obvodu bez poruchy s výstupem testovaného obvodu s poruchou. Princip metody je prostý, ale pro vyšší počet vstupních proměnných se tato metoda stává nepřehlednou a těžko použitelnou.

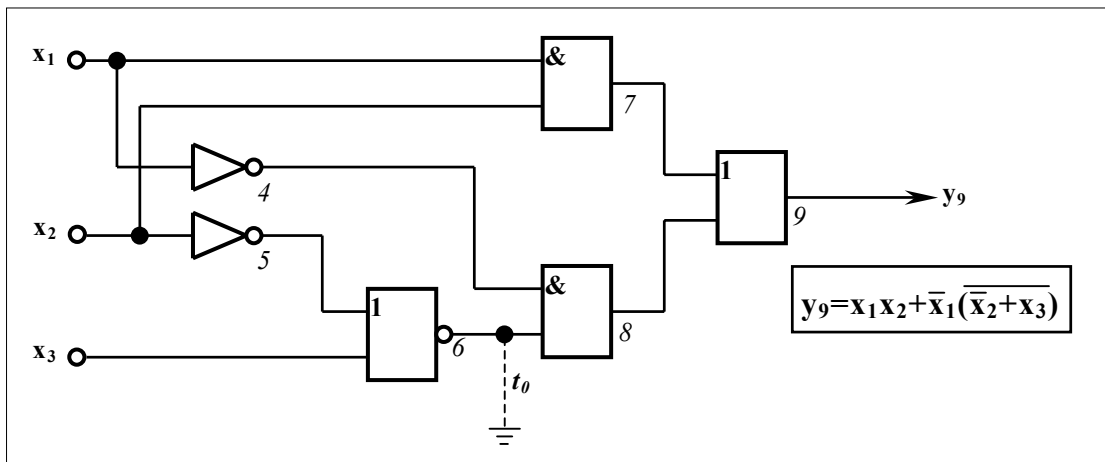
Testovacím vektorem konkrétní poruchy je taková vstupní kombinace testovaného obvodu pro kterou platí následující rovnice:

$$\mathbf{y}^* = \bar{\mathbf{y}}$$

Princip metody si ukážeme na konkrétních příkladech poruch kombinačního obvodu, který bude použit i pro znázornění postupu sestavení kroku testu ostatními metodami popisovanými v této kapitole.

Příklad 1.1:

Nalezněte tabulkovou metodou krok testu pro poruchu t_0 v uzlu 6 obvodu na obrázku č. 1.4.



Obr.č.1.4: Schéma obvodu s poruchou t_0 v uzlu 6.

Postup:

I. sestavení funkční tabulky obvodu s poruchou a obvodu bez poruchy. Funkce obvodu bez poruchy a její normalizované vyjádření je následující:

$$y = x_1x_2 + \bar{x}_1(\bar{x}_2 + x_3)$$

$$y = x_1x_2x_3 + x_1x_2\bar{x}_3 + \bar{x}_1x_2\bar{x}_3$$

Funkce obvodu s poruchou a její normalizované vyjádření je následující:

$$y^* = x_1x_2 + x_10$$

$$y^* = x_1x_2x_3 + x_1x_2\bar{x}_3$$

Tabulka výstupních hodnot obvodu bez poruchy a obvodu s poruchou je uvedena v následující tabulce. Stínováním je v tabulce vyznačen testovací vektor obvodu s poruchou t_0 v uzlu 6.

x_1	x_2	x_3	y	y^*
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

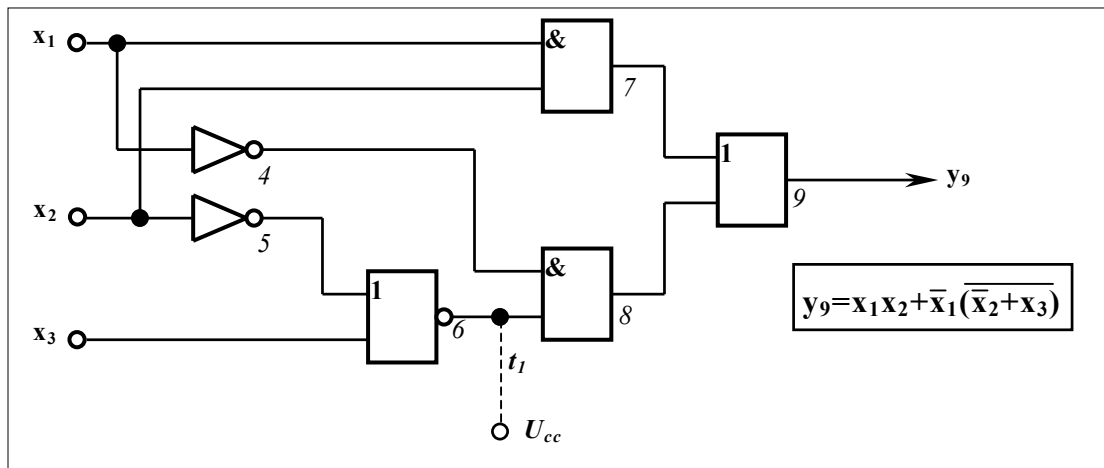
Řešení:

Testovací vektor poruchy t_0 v uzlu 6 je kombinace:

$$x_1, x_2, x_3 = 0, 1, 0 \text{ (ekvivalent 2)}$$

Příklad 1.2:

Nalezněte tabulkovou metodou krok testu pro poruchu t_1 v uzlu 6 obvodu na obrázku č. 1.5.



Obr.č.1.5: Schéma obvodu s poruchou t_1 v uzlu 6.

Postup:

I. sestavení funkční tabulky obvodu s poruchou a obvodu bez poruchy. Stínováním je v tabulce vyznačen testovací vektor obvodu s poruchou t_1 v uzlu 6.

Funkce obvodu bez poruchy a její normalizované vyjádření funkce je stejné jako u příkladu 1.1.

Funkce obvodu s poruchou a její normalizované vyjádření je následující:

$$y^* = x_1x_2 + \bar{x}_11$$

$$y^* = x_1x_2x_3 + x_1x_2\bar{x}_3 + \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3$$

Tabulka výstupních hodnot obvodu bez poruchy a obvodu s poruchou je uvedena v následující tabulce. Stínováním jsou v tabulce vyznačeny testovací vektory obvodu s poruchou t_1 v uzlu 6.

x_1	x_2	x_3	y	y^*
0	0	0	0	1
0	0	1	0	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

Řešení:

Podmínku pro testovací vektor poruchy t_1 v uzlu 6 splňují tři vstupní kombinace obvodu:

1. $x_1, x_2, x_3 = 0, 0, 0$ (ekvivalent 0)

2. $x_1, x_2, x_3 = 0, 0, 1$ (ekvivalent 1)

3. $x_1, x_2, x_3 = 0, 1, 1$ (ekvivalent 3)

1.3.1.3 Metoda citlivé cesty

Tato metoda vychází z podrobných znalostí fungování konstrukčních prvků systému a nevyžaduje komplexní znalosti o fungování systému jako celku. Pro její aplikaci je třeba mít k dispozici schéma obvodu, pomocí něhož hledáme podmínky pro demonstrování zvolené poruchy na výstupu obvodu.

Tuto metodu sestavení kroku testu popíšeme jako první. Logika konstrukce kroku je společná i pro ostatní metody. Pro některá zapojení nelze touto metodou nalézt testovací vektory i když existují.

Před vlastním sestavením kroku testu si pro jednoznačnou orientaci ve schéma obvodu označíme jednotlivé konstrukční prvky obvodu a jejich propojení. Propojení - vodiče - mezi prvky obvodu označujeme termínem *uzly*. Obvykle se označení volí tak, že vzestupně očíslovujeme vstupy obvodu a poté pokračujeme v číslování jednotlivých prvků obvodu, tak, že směrem k výstupu obvodu číselná označení rostou. Pro označení uzlů obvodu použijeme stejné hodnoty jako pro konstrukční prvky připojené k tomuto uzlu výstupem (pokud je obvod bez závady, je k uzlu v definovaném čase svým výstupem připojen vždy pouze jeden aktivní konstrukční prvek). Při dodržení tohoto postupu dosáhneme toho, že uzly připojené ke vstupům konstrukčních prvků mají nižší číselné označení než jejich výstupy. Tento způsob označení uzlů a prvků systému usnadňuje orientaci ve schéma obvodu i pro další metody definice kroku testu.

Postup sestavení kroku testu metodou citlivé cesty:

- I. část sestavení kroku testu - zpětné šíření poruchy
 - a) do místa poruchy vložíme inverzní hodnotu - toto je podmínka demonstrace poruchy,
 - b) vyšetřujeme jaké logické hodnoty musí být na vstupech logického prvku, aby byla splněna podmínka předchozího bodu,
 - c) získané logické hodnoty jsou podmínkou pro nalezení hodnot vstupních signálů předchozí vrstvy logických prvků. Musíme dbát na zachování kontinuity signálů ve všech vrstvách logické sítě - pro nastavení určité logické hodnoty na výstupu je možno použít více vstupních kombinací na vstupech příslušného logického prvku, ale v závislosti na schéma testovaného obvodu mohou být dále použity pouze některé z nich. Všechny vyhovující kombinace vstupních signálů logického prvku - *realizace* - musí být zohledněny pro hledání akceptovatelných logických hodnot ve vrstvách logické sítě bližších vstupům testovaného obvodu,

- d) podle předchozího bodu pokračujeme až ke vstupům testovaného obvodu,
- e) protínáním realizací zpětného šíření (protínání realizací znamená, že na spojích logických obvodů může být vždy pouze jedna hodnota z n případně možných) vybíráme použitelné a bezkonfliktně nastavitelné logické hodnoty, dokud nenalezneme všechny vstupní kombinace testovaného obvodu, které v místě poruch nastavují inverzní hodnotu,

II. část sestavení kroku testu - otevření cesty na výstup

- a) nyní hledáme takové logické hodnoty na vstupech následujících logických prvků, které umožní šíření chybového signálu z místa poruchy na výstup testovaného obvodu. Získané logické hodnoty jsou podmínkou pro nalezení hodnot vstupních signálů logických prvků otevírajících cestu chybového signálu na výstup,
- b) z hodnot vstupních logických prvků otevírajících cestu chybového signálu na výstup odvodíme hodnoty vstupních signálů předchozí vrstvy logických prvků. Pro nastavení určité logické hodnoty na výstupu logického prvku je možno použít více vstupních kombinací na vstupech příslušného logického prvku, ale v závislosti na schéma obvodu mohou být dále použity pouze některé z nich,
- c) podle předchozího bodu pokračujeme až ke vstupům testovaného obvodu,
- d) opět musíme dbát na zachování kontinuity signálů ve všech vrstvách logické sítě. Všechny vyhovující realizace musí být zohledněny pro hledání akceptovatelných logických hodnot ve vrstvách logické sítě bližších výstupům testovaného obvodu,

III. část sestavení kroku testu - definice vstupních vektorů (kroků testu)

- a) protnutím výsledků zpětného šíření a otevření cesty na výstup dostaneme vyhovující vstupní kombinace - kroky testu.

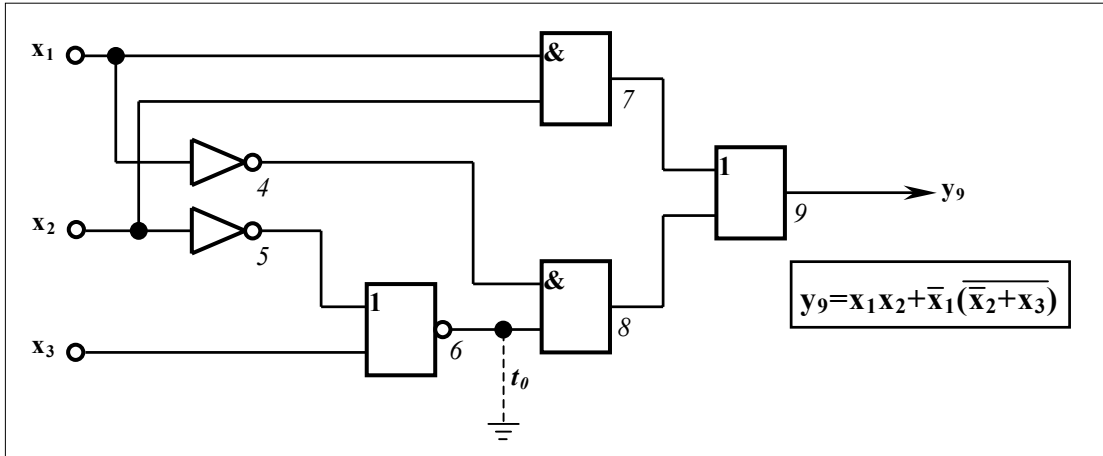
Celý postup si znázorníme v následujícím příkladu obvodu se základními logickými obvody typu invertor, negovaný logický součet, přímý logický součet a přímý logický součin.

Pro názornost bude pro každou dílčí část sestavení kroku testu vkládána do schéma obvodu k označení uzlu za rovnítko vhodná logická úroveň. Použitelná hodnota bude na dalším obrázku označena tučným písmem. Pokud splňuje podmínky více kombinací vstupních signálů, za rovnítkem může být uvedeno více hodnot oddělených navzájem čárkou. Pro hledání všech vyhovujících kombinací je třeba používat stejnohlé hodnoty (podle pořadí za rovnítkem).

Konkrétní postup se může značně lišit podle místa a typu modelu poruchy.

Příklad 1.3:

Nalezněte metodou citlivé cesty krok testu pro poruchu t_0 v uzlu 6 obvodu na obrázku č. 1.6. Poruchu můžeme zobrazit jako zkrat uzlu 6 na zemní vodič.

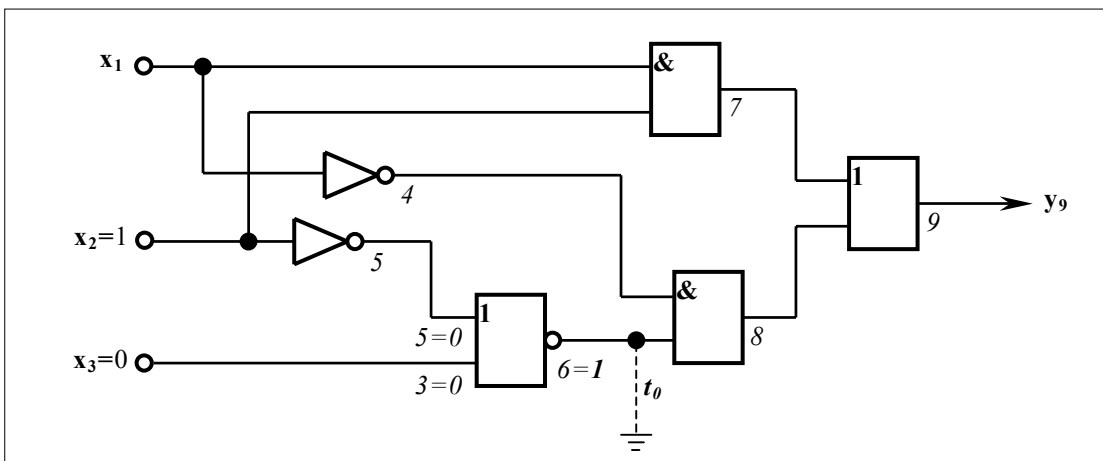


Obr.č.1.6: Schéma obvodu s poruchou t_0 v uzlu 6.

Postup:

I. zpětné šíření poruchy

- do uzlu 6 vložíme hodnotu logická 1,
- na vstupu hradla 6 (negovaný součet) musí být v uzlech 5 a 3 hodnoty **log.0**,
- na vstupu hradla 5 (invertor) musí být na uzlu 2 hodnota **log.1**,

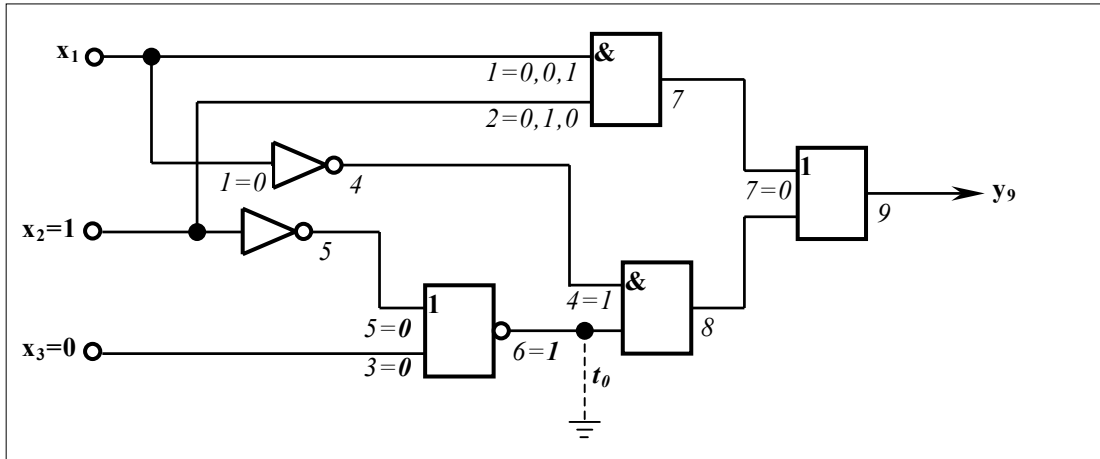


Obr.č.1.6.1: Výsledek první části sestavení kroku testu metodou citlivé cesty.

II. otevření cesty na výstup

- Nyní známe jaké musí být hodnoty v uzlech 2, 3 a 5, aby v místě poruchy - uzel 6 - byla nastavena podmínka demonstrace poruchy,
- aby demonstrováná porucha mohla být pozorovatelná na výstupu, musí být pro uzel 6 otevřena hradla 8 (přímý součin) a 9 (přímý součet),

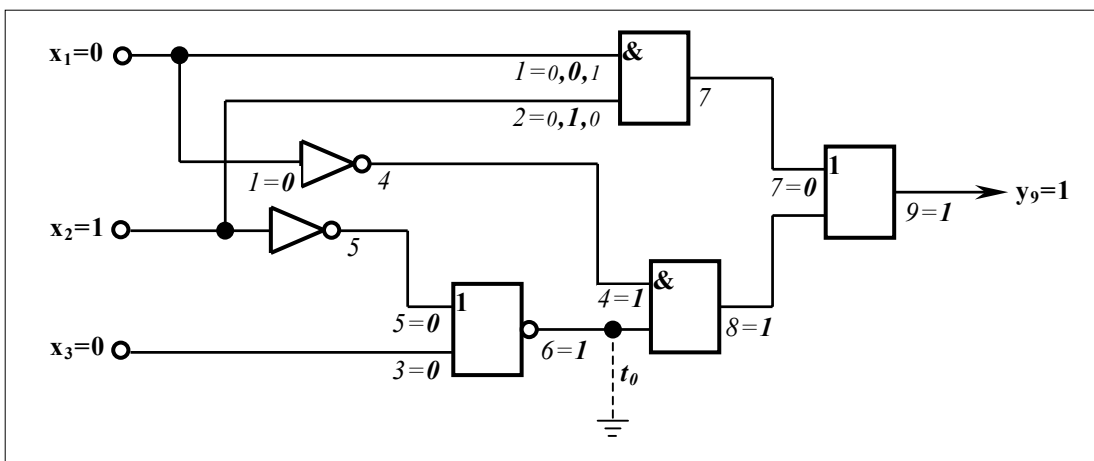
- b) v uzlu 4 (invertor) musí být **log.1** a v uzlu 7 (přímý součin) musí být **log.0**,
 c) pokud v uzlu 4 musí být **log.1**, pak v uzlu 1 musí být **log.0**,
 d) pokud v uzlu 7 má být **log.0**, pak v uzlu 1 nebo 2, případně v obou musí být **log.0**,



Obr.č.1.6.2: Výsledek druhé části sestavení kroku testu metodou citlivé cesty.

III. definice vstupních vektorů, kroků testu

- a) Dosavadním postupem jsme získali následující vstupní hodnoty:
 $3 = 0, 2 = 1, 1 = 0; 1 = 0, 2 = 1; 1 = 1, 2 = 0$ a $1 = 2 = 0$
 b) výsledek získáme logickým průnikem těchto hodnot.



Obr. č.1.6.3: Výsledek třetí části sestavení kroku testu metodou citlivé cesty.

Řešení:

Na každém spoji může být pouze jedna logická hodnota. Testovací vektor poruchy t_0 v uzlu 6 je kombinace:

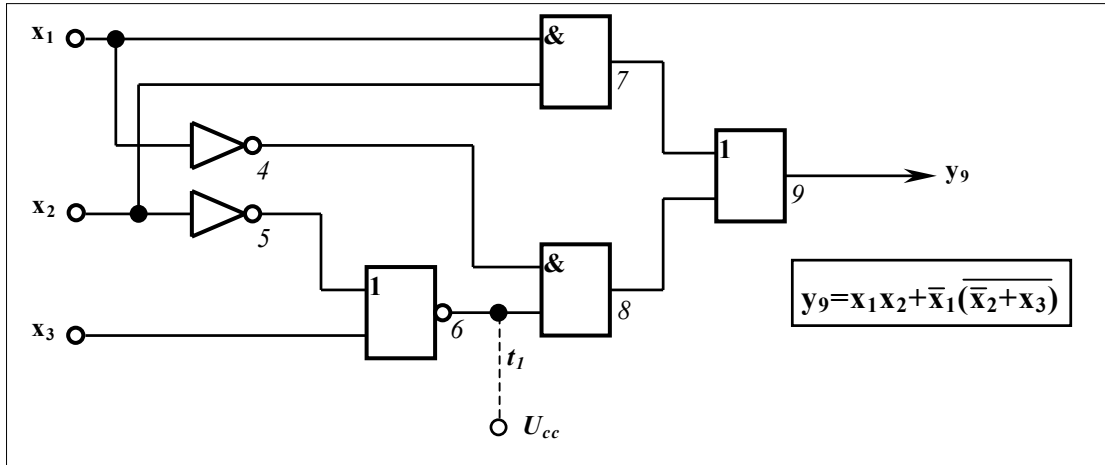
$$x_1, x_2, x_3 = 0, 1, 0 \text{ (ekvivalent 2)}$$

Příklad 1.3 je jednodušší variantou realizace zpětného protínání. Často je třeba zvažovat větší počet vstupních kombinací vyhovujících

realizaci poruchy. Pro přehledné zvládnutí této komplikace je vhodné používat již popsany postup protínání stejnohlých realizací vstupních kombinací viz příklad 1.4.

Příklad 1.4:

Nalezněte krok testu pro poruchu t_1 v uzlu 6 obvodu viz. obr. č. 1.7.

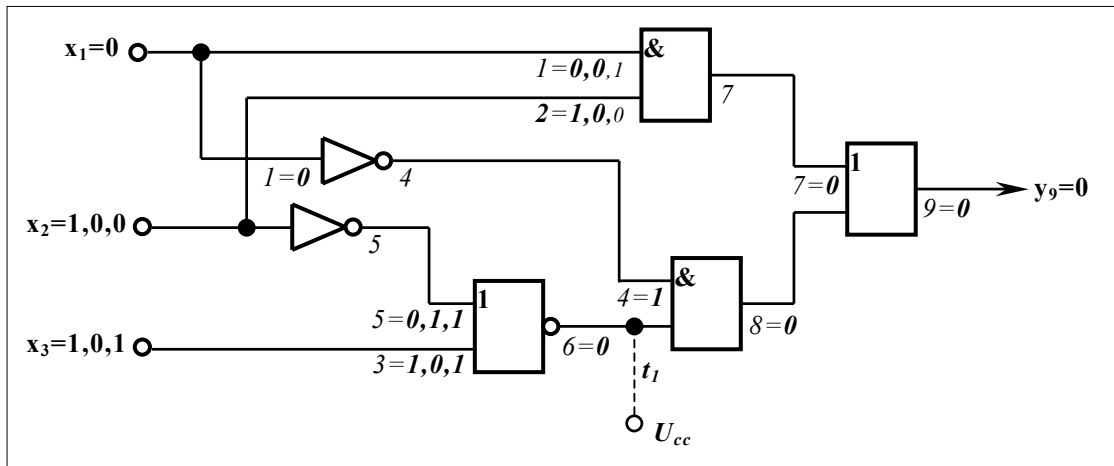


Obr.č.1.7: Schéma obvodu s poruchou t_1 v uzlu 6.

Porucha obvodu je zobrazena jako připojení uzlu 6 na napájecí napětí obvodu. Podstatné je, že obvod 8 pracuje s **log.1** na vstupu 6.

Postup:

Postup je obdobný příkladu 1.1, proto bude zobrazen přímo výsledek postupu, viz obr. č.1.7.1.



Obr.č.1.7.1: Výsledek sestavení kroku testu metodou citlivé cesty

Řešení:

Právě třetí etapa sestavení kroku testu přináší určité komplikace. Lze vypořadovat, že hodnoty v uzlu 5 jsou vázány podmínkami v uzlu 3 a zároveň je v závislosti na podmínce v uzlu 1 (vstup hradla 7) možno na uzel 2 přiložit libovolnou hodnotu. Je zřejmé, že protínání realizací vyžaduje znalost fungování konstrukčních prvků a nelze ho v této

podobě nijak snadno algoritmovat.

Podmínku pro testovací vektor poruchy t_1 v uzlu 6 splňují tři vstupní kombinace obvodu:

1. $x_1, x_2, x_3 = 0, 1, 1$ (ekvivalent 3)

2. $x_1, x_2, x_3 = 0, 0, 0$ (ekvivalent 0)

3. $x_1, x_2, x_3 = 0, 0, 1$ (ekvivalent 1)

1.3.1.4 Metoda Boolovské difference

Pro eliminaci problémů s vyhodnocením protínání realizací slouží tzv. Boolovská difference. Tato metoda představuje analytický přístup využívající pravidla Boolovy algebry pro popis metody citlivé cesty.

Před vlastním sestavením kroku testu si jednoznačně označíme, obdobně jako v předchozí metodě, především vstupní a výstupní uzly testovaného obvodu.

Při konstrukci kroku testu vycházíme z logické funkce, kterou testovaný obvod plní. Logickou funkci odvodíme ze schéma testovaného obvodu a neprovádíme její minimalizaci. Tímto opatřením zajistíme generování kroků strukturního testu. Boolovská difference tedy umožňuje navrhovat kroky testu redundantních obvodů.

Pokud bychom přistoupili na minimalizaci logické funkce realizované testovaným obvodem, budeme generovat kroky funkčního testu. Funkční test sice může posloužit k prověření správné funkce obvodu, ale neumožňuje provádět přesnou lokalizaci poruchy.

Postup sestavení kroku testu metodou Boolovské difference:

- I. část sestavení kroku testu - substituce poruchy
 - a) sestavíme logickou funkci části obvodu realizující dílčí funkci v místě poruchy, kterou označíme symbolem h .
 $h(x_i, x_j, \dots, x_n)$
kde x_i, x_j, \dots, x_n jsou vstupní signály testovaného obvodu podílející se na vytváření logické hodnoty v místě poruchy testovaného obvodu,
- II. část sestavení kroku testu - vyjádření Boolovské difference
 - a) pomocí funkce h sestavíme modifikovanou rovnici testovaného obvodu - provedeme *substituci* poruchy do testovaného obvodu:
 $y = F(x_1, x_2, \dots, x_z, h)$
kde x_i, x_j, \dots, x_z jsou vstupní signály testovaného obvodu,
 - b) sestavení logické funkce otevření cesty na výstup testovaného obvodu.

Sestavení logické funkce otevření cesty na výstup vychází z požadavku splnění podmínky demonstrace poruchy na výstupu

obvodu. Vstupní hodnoty testovaného obvodu musí být nastaveny tak, aby změny logické úrovně v místě poruchy nebyly maskovány, tedy aby byly přenášeny na výstup bez jakéhokoliv zkreslení, což jinak řečeno znamená otevření cesty z místa poruchy na výstup. Každá změna logické hodnoty v místě poruchy se musí projevit na výstupu.

Pokud je cesta z místa poruchy na výstup testovaného obvodu otevřena, pak platí následující vztahy, které představují upravené modifikované logické funkce testovaného obvodu:

$$\begin{aligned} y &= F(x_1, x_2, \dots, x_z, 0) \\ \bar{y} &= F(x_1, x_2, \dots, x_z, 1) \end{aligned}$$

Pro nalezení hodnot vstupních vektorů (což je vlastně množina hodnot vstupních proměnných x_1, x_2, \dots, x_z) je třeba najít logickou funkci, která umožní jednoznačně stanovit podmínky pro určení jejich hodnoty. Pro sestavení této funkce se jako velmi vhodná jeví logická rovnice využívající funkce **XOR** (součet modulo 2), do níž dosadíme upravené modifikované logické funkce testovaného obvodu. Tato funkce je v literatuře nazývána **Boolovskou diferencí**.

První upravenou modifikovanou logickou funkci obvodu získáme tak, že do modifikované logické funkce obvodu dosadíme za proměnnou **h** hodnotu **log.0**. Druhou upravenou modifikovanou logickou funkci obvodu získáme obdobným způsobem, do téže modifikované logické funkce dosadíme za proměnnou **h** hodnotu **log.1**.

Boolovská diference obvodu má následující tvar:

$$dy/dh = [y] \text{ XOR } [\bar{y}]$$

Pokud dosadíme za **y** upravené modifikované výstupní funkce dostáváme následující vztah pro Boolovskou diferenci obvodu s poruchou:

$$dF/dh = [F(x_1, x_2, \dots, x_z, 0)] \text{ XOR } [F(x_1, x_2, \dots, x_z, 1)]$$

Tato logická funkce jednoznačně popisuje podmínky pro stanovení vstupních vektorů testovaného obvodu otevírajících cestu z místa poruchy na výstup obvodu. Dostáváme tak hodnoty vstupních proměnných testovaného obvodu, které umožňují demonstraci poruchy na výstupu.

III. část sestavení kroku testu - zpětné šíření poruchy

- a) zpětné šíření poruchy znamená určení hodnoty vstupních signálů, které v místě poruchy nastaví opačnou logickou úroveň, než kterou je reprezentována porucha,

- b) logickou funkci v místě poruchy rovnici jsme již určili v části I.a) jako funkce **h**, která v této části byla použita pro substituci poruchy do testovaného obvodu,
- c) pro model poruchy typu t_0 použijeme přímý tvar funkce **h**. Běžné vyjádření logické funkce představuje zápis, kdy tato funkce nabývá hodnoty **log.1**.
- d) pro model poruchy typu t_1 použijeme inverzní tvar funkce \bar{h} .

IV. část sestavení kroku testu - definice vstupních vektorů

- a) obdobně jako v metodě citlivé cesty je nutno provést protnutí vstupních vektorů otevírajících cestu z místa poruchy na výstup a vstupních vektorů zajišťujících zpětné šíření poruchy,
- b) krokem testu může být pouze taková kombinace vektorů, která umožňuje zároveň jak otevření cesty z místa poruchy na výstup, tak zpětné šíření poruchy. K protnutí vektorů, pro vyjádření výše uvedené vlastnosti, slouží funkce **AND**,
- c) kroky testu pro model poruchy typu t_0 získáme řešením logické rovnice:

$$[\mathbf{h}] \text{ AND } [\mathbf{dF/dh}] = 1$$
- d) kroky testu pro model poruchy typu t_1 získáme řešením následující logické rovnice:

$$[\bar{\mathbf{h}}] \text{ AND } [\mathbf{dF/dh}] = 1$$
- e) Vyřešením těchto rovnic dostaneme vstupní vektory představující kroky testu prověřovaného obvodu.

Boolovská diference představuje účinný analytický nástroj pro hledání kroku testu kombinačních obvodů. Je třeba si uvědomit, že se nejedná o klasickou diferenci v matematickém slova smyslu. Postup získání kroku testu bude vysvětlen na příkladě č. 1.5.

Příklad 1.5:

Nalezněte metodou Boolovské diference krok testu pro poruchu t_0 a t_1 v uzlu 6 obvodu na obrázku č. 1.8. Poruchu t_0 můžeme zobrazit jako zkrat uzlu 6 na zemní vodič a poruchu t_1 jako připojení uzlu 6 na napájecí napětí obvodu.

Postup:

I. substituce poruchy

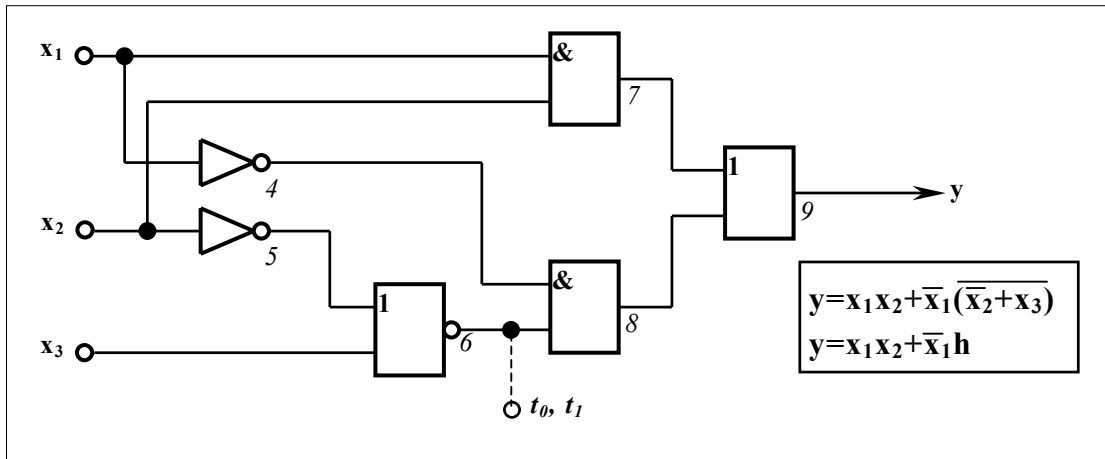
*funkci **h** plní část obvodu včetně hradel 6 a 5. Po substituci má funkce testovaného obvodu tvar:*

$$y = x_1 x_2 + \bar{x}_1 h$$

kde

$$h = (\bar{x}_2 + x_3)$$

S takto definovanou substitucí budeme pracovat ve všech fázích odvození kroku testu.



Obr.č.1.8: Schéma obvodu s poruchami t_0 a t_1 v uzlu 6.

II. vyjádření Boolovské diference

Následující řádky budou demonstrovat rozvoj modifikovaných funkcí testovaného obvodu a vyjádření funkce XOR základními logickými funkcemi.

$$dF/dh = [F(x_1, x_2, \dots, x_z, 0)] \text{ XOR } [F(x_1, x_2, \dots, x_z, 1)]$$

$$dF/dh = [x_1x_2 + \bar{x}_10] \text{ XOR } [x_1x_2 + \bar{x}_11]$$

$$dF/dh = [x_1x_2] \text{ XOR } [x_1x_2 + \bar{x}_1]$$

$$dF/dh = (\overline{x_1x_2}) (x_1x_2 + \bar{x}_1) + (x_1x_2) (\overline{x_1x_2 + \bar{x}_1})$$

$$dF/dh = (\overline{x_1x_2}) \bar{x}_1 + (x_1x_2) (\bar{x}_1\bar{x}_2x_1)$$

$$dF/dh = (\bar{x}_1 + \bar{x}_2) \bar{x}_1 + (x_1x_2) (\bar{x}_1 + \bar{x}_2)x_1$$

$$dF/dh = \bar{x}_1 + \bar{x}_2\bar{x}_1 + x_1x_2\bar{x}_1x_1 + x_1x_2\bar{x}_2x_1$$

$$dF/dh = \bar{x}_1 + \bar{x}_2\bar{x}_1$$

$$dF/dh = \bar{x}_1(1 + \bar{x}_2)$$

$$dF/dh = \bar{x}_1$$

III. definice vstupních vektorů - kroků testu

a) kroky testu pro model poruchy typu t_0 :

$$[h] \text{ AND } [dF/dh] = 1$$

$$[\overline{x_2 + x_3}] \text{ AND } [x_1] = 1$$

$$\bar{x}_1x_2\bar{x}_3 = 1$$

Řešení:

Testovací vektor poruchu t_0 v uzlu 6 je kombinace:

$$x_1, x_2, x_3 = 0, 1, 0 \text{ (ekvivalent 2)}$$

b) kroky testu pro model poruchu typu t_1 získáme řešením následující logické rovnice:

$$[\bar{h}] \text{ AND } [dF/dh] = 1$$

$$[\bar{x}_2 + x_3] \text{ AND } [\bar{x}_1] = 1$$

$$\bar{x}_1 \bar{x}_2 + \bar{x}_1 x_3 = 1$$

$$\bar{x}_1 \bar{x}_2 (x_3 + \bar{x}_3) + \bar{x}_1 (x_2 + \bar{x}_2) x_3 = 1$$

$$\bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + \bar{x}_1 \bar{x}_2 x_3 = 1$$

$$\bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 = 1$$

Řešení:

Podmínku pro testovací vektor poruchy t_1 v uzlu 6 splňují tři vstupní kombinace obvodu:

1. $x_1, x_2, x_3 = 0, 0, 0$ (ekvivalent 0)

2. $x_1, x_2, x_3 = 0, 0, 1$ (ekvivalent 1)

3. $x_1, x_2, x_3 = 0, 1, 1$ (ekvivalent 3)

Dostáváme tedy stejné výsledky jako v předchozích příkladech, Postup řešení je poněkud oprostěn od konkrétní technické realizace. Naopak je orientován na formální postup řešení logických vztahů, poskytujících exaktní popis testovaného obvodu.

Je nutno zdůraznit, že Boolovská diference není derivací, i když se takto nazývá. Je to označení pro speciální logickou operaci nad logickou rovnicí obvodu, která vyjadřuje vliv substituce na výstupní hodnotu logické funkce.

Boolovská diference má několik zajímavých vlastností. Popíšeme si je pro logickou funkci ve tvaru:

$$y = F(x_1, x_2, \dots, x_i, \dots, x_z)$$

- pro její vyjádření platí následující vztah

$$dF/dx_i = [F(x_1, x_2, \dots, 0, \dots, x_z)] \text{ XOR } [F(x_1, x_2, \dots, 1, \dots, x_z)]$$

- v tomto vztahu je jednou za x_i dosazena **log. 0** a podruhé **log. 1**,
- jestliže $dF/dx_i = 0$, pak je Boolovská diference nezávislá na x_i ,
- naopak pokud $dF/dx_i = 1$, je Boolovská diference závislá na x_i a změna x_i způsobí změnu hodnoty výstupu,
- obecně může být Boolovská diference funkcí všech vstupních proměnných mimo x_i nebo jejich podmnožinou,
- Boolovskou diferenci lze vyjádřit pro neminimalizované funkce.

Pro rozvoj a zjednodušení vyjádření složitých a rozsáhlých vztahů pro Boolovskou diferenci lze použít následující pravidla:

$$\overline{dF/dx_i} = dF/dx_i$$

$$dF/dx_i = dF/d\overline{x_i}$$

$$d(dF/dx_j)/dx_i = d(dF/dx_i)/dx_j$$

$$d([F] \text{ XOR } [G])/dx_i = [dF/dx_i] \text{ XOR } [dG/dx_i]$$

$$d([F] \text{ AND } [G])/dx_i = ([F] \text{ AND } [dG/dx_i]) \text{ XOR } ([dF/dx_i] \text{ AND } [G]) \\ \text{ XOR } ([dF/dx_i] \text{ AND } [dG/dx_i])$$

$$d([F] \text{ OR } [G])/dx_i = ([\overline{F}] \text{ AND } [dG/dx_i]) \text{ XOR } ([dF/dx_i] \text{ AND } [\overline{G}]) \\ \text{ XOR } ([dF/dx_i] \text{ AND } [dG/dx_i])$$

1.3.1.5 Metoda D-algoritmu

Z dosud popsaných metod je zřejmé, že sice poskytují představu o mechanismu sestavení kroku testu, ale na jejich základě nelze snadno sestavit algoritmus pro nalezení kroku testu. Tento nedostatek odstraňuje metoda založená na tzv. D-algoritmu. D-algoritmus byl popsán řádově před 30 lety, ale poskytuje schůdný aparát pro automatizované generování kroků testů. Soudobé návrhové systémy číslicových systémů obsahují jako svou nedílnou součást také úlohu na sestavení testu navrhovaného obvodu, vycházející z mnoha modifikací D-algoritmu. Požadavek na vestavení funkce sestavení testu navrhovaného systému vyplývá z požadavků řady oborů, kde výrobce a dodavatel je povinen připravit kompletní podklady pro detekci a lokalizaci poruch dodávaného systému.

Ve své podstatě D-algoritmus opět vychází z metody citlivé cesty. Používá stejný způsob číslování uzlů. Pro nalezení kroku testu používá specifický mechanismus tabulek umožňující snadnou algoritmizaci.

Nalezení kroku testu je oproštěno od analytického řešení. Tato metoda vyžaduje mechanické splnění všech fází řešení a prověření všech možností, které konkrétní zapojení obvodu poskytuje. Jak již bylo řečeno, jedná se o algoritmus, který prověřuje všechny možné varianty, i ty, které jsou pro řešitele nadaného znalostmi z konstrukce číslicových obvodů předem zcela bezpředmětné.

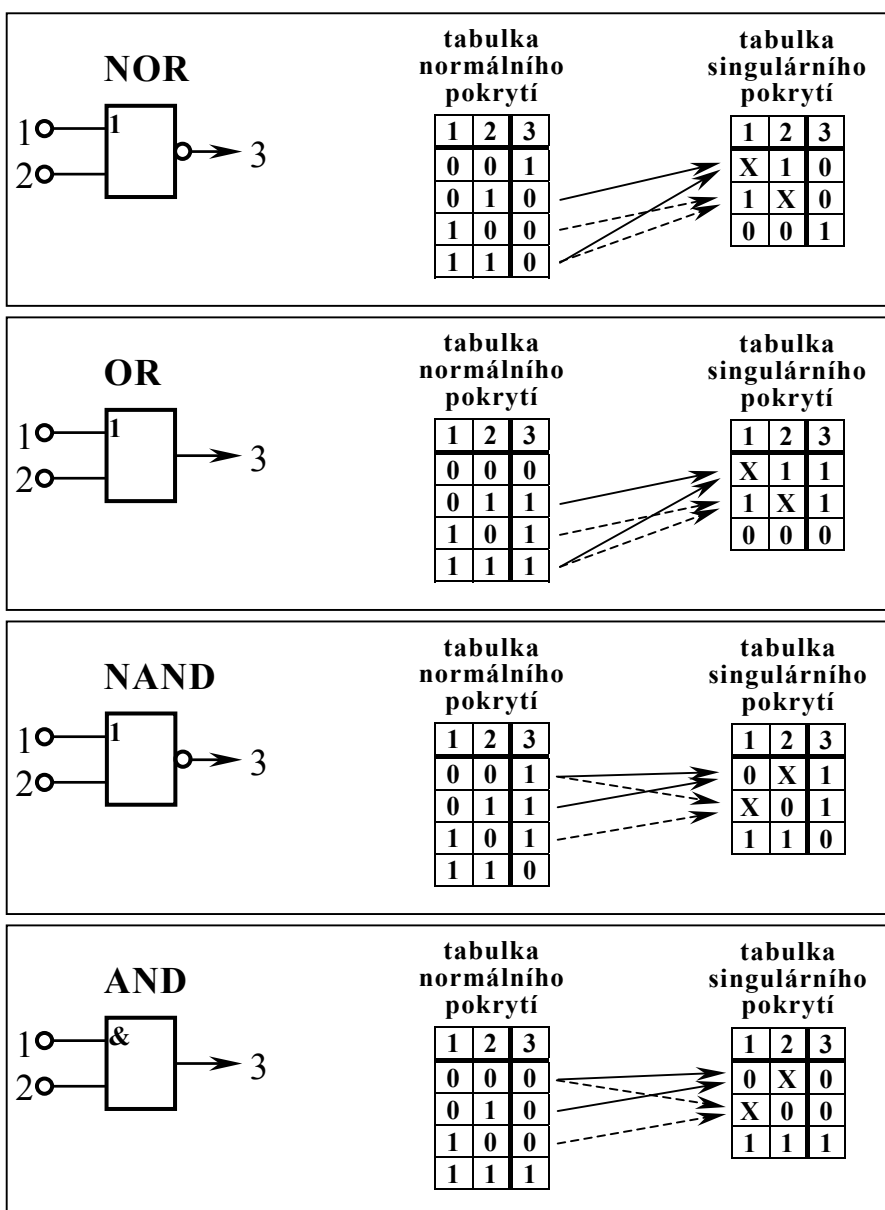
Postup sestavení kroku testu D-algoritmem:

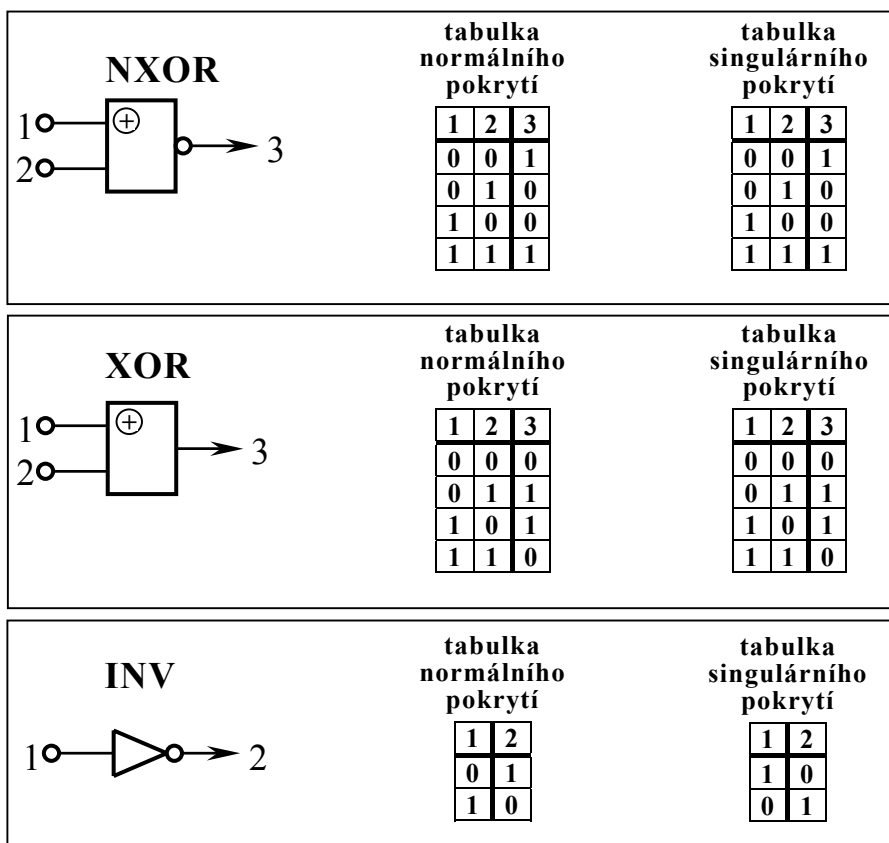
I. část sestavení kroku testu - konstrukce tabulek **singulárního pokrytí** konstrukčních prvků testovaného obvodu

a) tabulka singulárního pokrytí se vytváří z tabulky normálního pokrytí konstrukčního prvku testovaného obvodu. Tabulka singulárního pokrytí vyjadřuje situaci zablokování průchodu signálů konstrukčním prvkem, tedy nastavení konstantní logické úrovně na výstupu konstrukčního prvku. Jedná se o dílčí redukci tabulky normálního pokrytí. Tato tabulka je pochopitelně

univerzálně použitelná a nemusí se opakovaně sestavovat pro každý další testovaný obvod. Ze singulárních tabulek všech použitých konstrukčních prvků se sestavuje tabulka singulárního pokrytí testovaného obvodu,

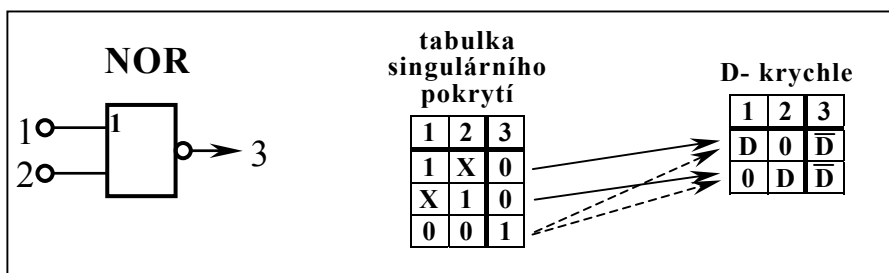
- b) formálně lze tento proces popsat jako protínání řádků tabulky normálního pokrytí, které mají stejnou výstupní hodnotu a liší se pouze jedním vstupním signálem. Totožné hodnoty na vstupu zůstávají zachovány, rozdílné vstupní hodnoty nahradíme symbolem **X**,
- c) tabulky singulárního pokrytí pro všechny běžné konstrukční prvky jsou uvedeny níže:

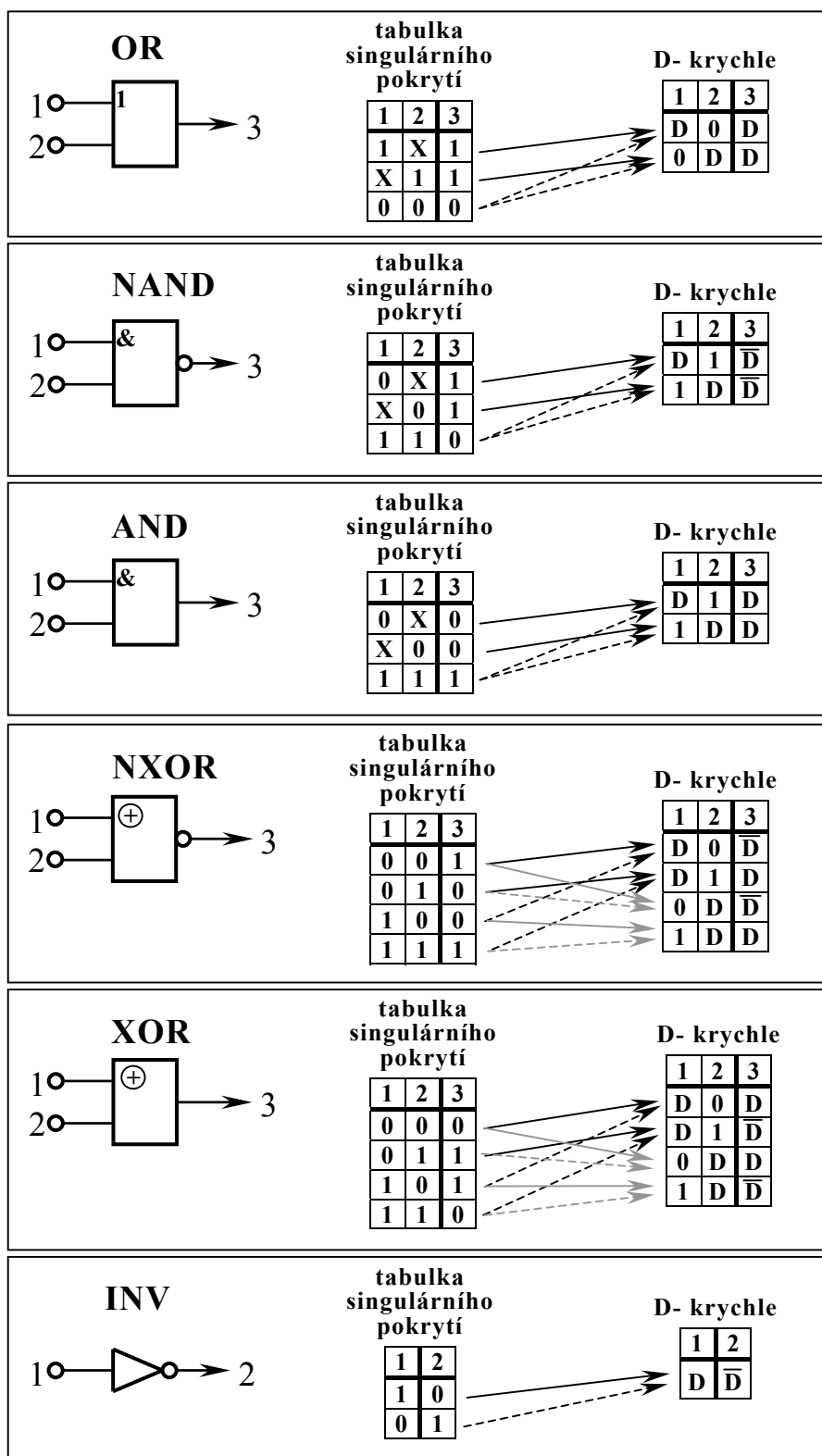




II. část sestavení kroku testu - konstrukce **D-krychle** konstrukčních prvků testovaného obvodu

- a) D-krychle představuje další redukci tabulky normálního pokrytí konstrukčního prvku testovaného obvodu. D-krychle znázorňuje situaci zprůchodnění konstrukčního prvku pro signál na jednom ze vstupů konstrukčního prvku. D-krychle konstrukčního prvku je také univerzálně použitelná a nemusí se opakovaně sestavovat pro každý testovaný obvod. Z D-krychlí všech použitých konstrukčních prvků se sestavuje D-krychle testovaného obvodu,
- b) formálně lze tento proces popsat jako protínání řádků tabulky singulárního pokrytí se symbolem **X** s řádkem bez symbolu **X**. Reálně D-krychle uvádí nastavení vstupních signálů umožňujících průchod signálu konstrukčním prvkem. Signál se označuje symbolem **D**,
- c) D-krychle všech běžných konstrukčních prvků jsou uvedeny níže:



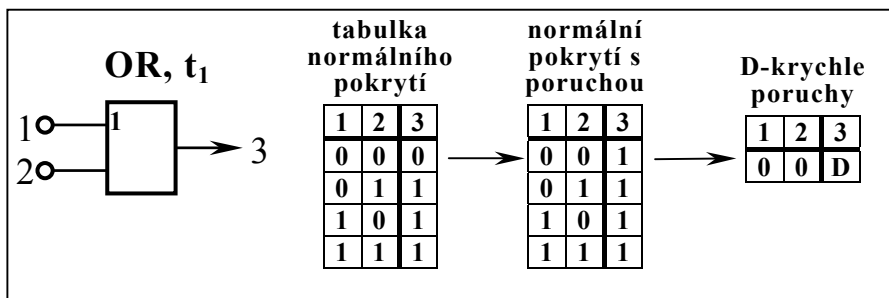
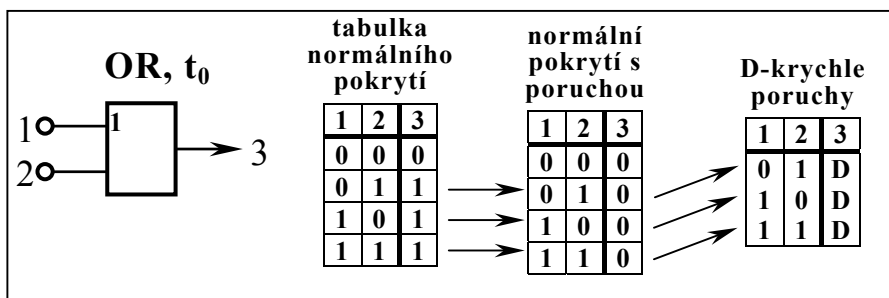
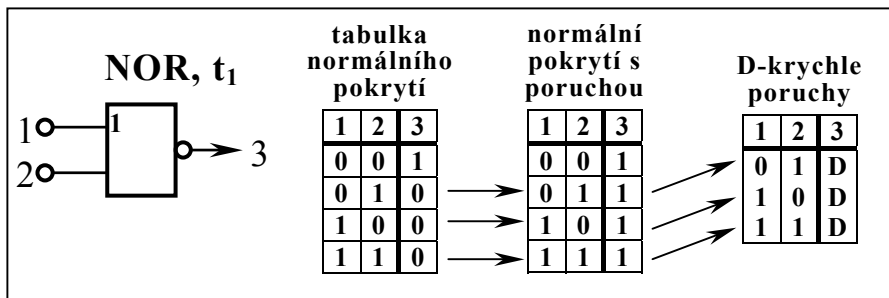
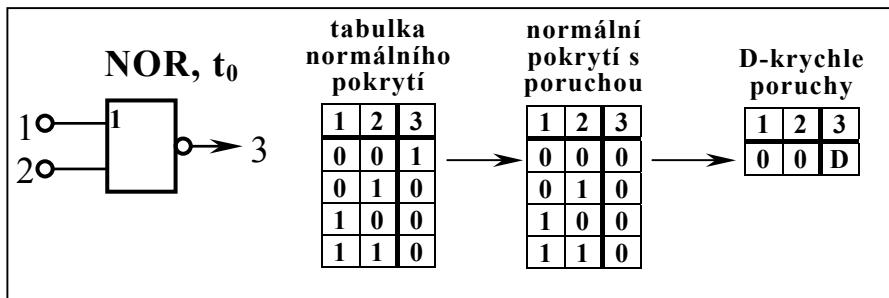


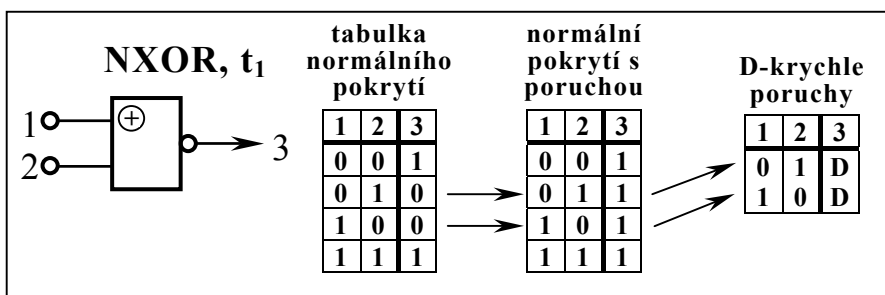
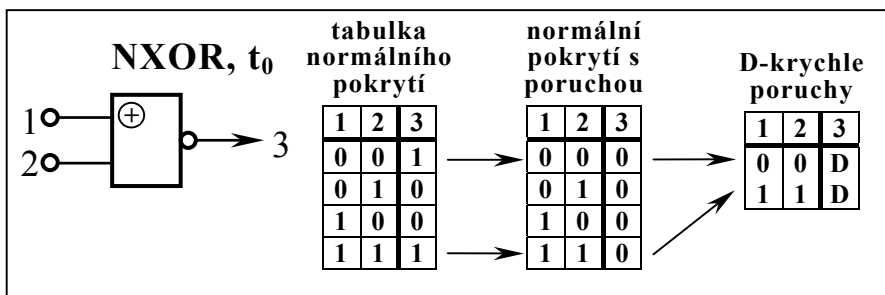
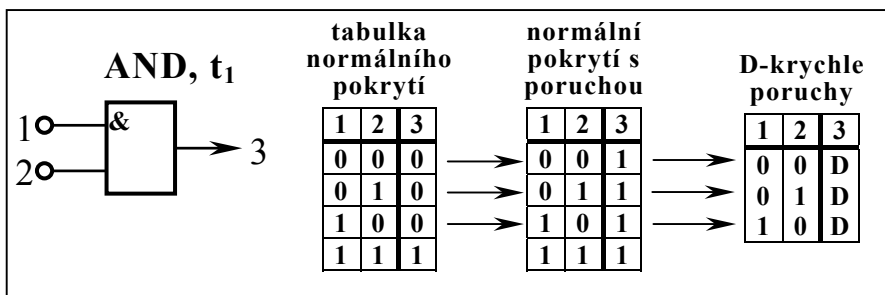
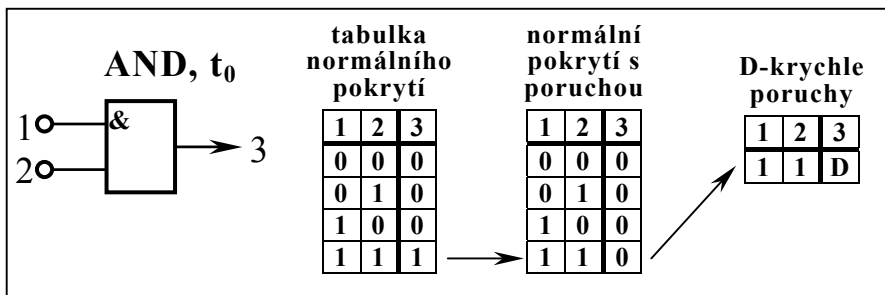
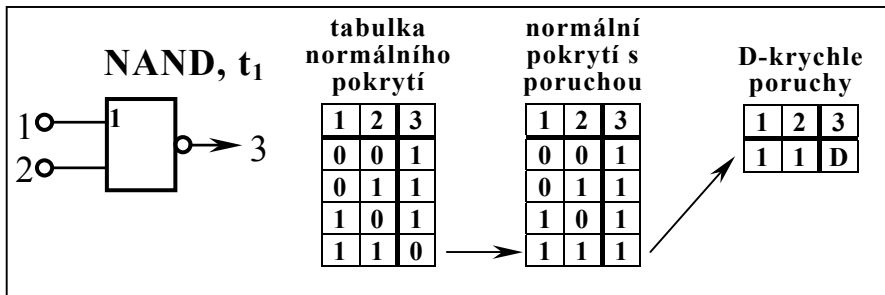
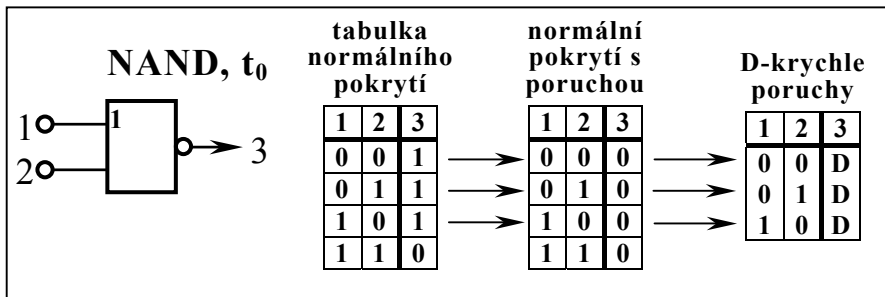
III. část sestavení kroku testu - konstrukce **D-krychle poruchy**

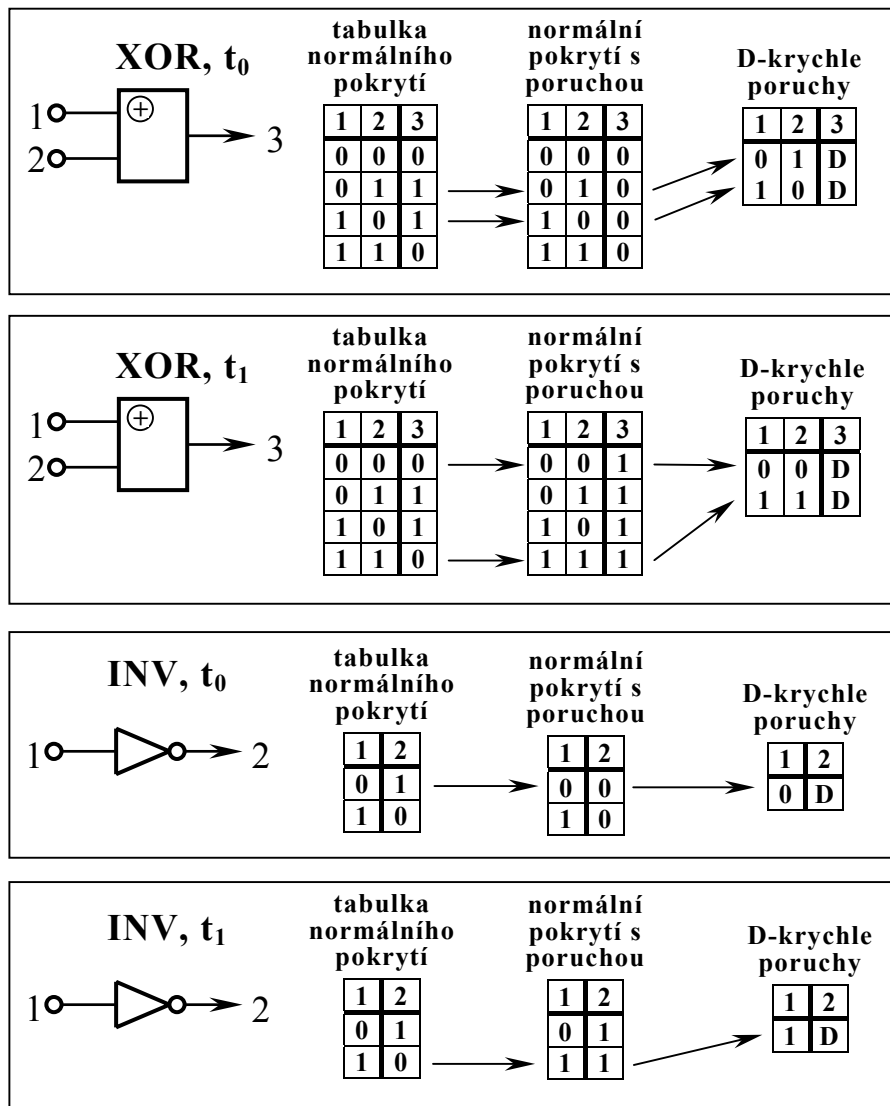
- porucha se v testovaném obvodu projevuje změnou logické hodnoty v uzlu k němuž je připojen konstrukční prvek s poruchou,
- D-krychli poruchy vytvoříme z řádku tabulky normálního pokrytí konstrukčního prvku bez poruchy a ze stejnohléhlého řádku tabulky normálního pokrytí konstrukčního prvku s poruchou. V těchto

tabulkách se uvažují pouze řádky v nichž je zapsána inverzní hodnota výstupu,

- c) výsledkem je tabulka u níž jsou na vstupu totožné hodnoty a na výstupu je díky projevu poruchy změna logické hodnoty. Tato změna je indikována symbolem **D**. Lze použít inverzní symbol, ale to není podstatné. Zásadním požadavkem je dodržet smysl změny během celého procesu odvození kroku testu. Symbol **D** znamená buď změnu z **log.0** do **log.1** nebo naopak z **log.1** do **log.0**,
- d) konkrétní D-krychle poruch jednotlivých konstrukčních prvků jsou následující:







IV. část sestavení kroku testu - konstrukce tabulky **singulárního pokrytí** testovaného obvodu

- vzestupně očíslovíme uzly testovaného obvodu, přičemž dodržíme pravidlo vzestupnosti - výstup konstrukčního prvku testovaného obvodu musí mít vždy přiřazeno nejvyšší číslo. Takto si zajistíme snadnou orientaci v tabulkách při hledání kroku testu,
- připravíme si tabulku s tolika sloupci, kolik má testovaný obvod uzlů,
- do tabulky vpisujeme postupně řádky tabulek singulárního pokrytí konstrukčních prvků, přičemž v souladu se schéma testovaného obvodu ztotožňujeme čísla vývodů konstrukčních prvků s čísly sloupců tabulky,
- ke každému sloupci tabulky singulárního pokrytí testovaného obvodu musí být přiřazena minimálně jedna výstupní špička konstrukčního prvku jako zdroj signálu pro tento uzel a minimálně jedna vstupní špička následujícího konstrukčního prvku testovaného obvodu,

V. část sestavení kroku testu - konstrukce tabulky **D-pokrytí** testovaného obvodu

- a) připravíme si tabulku s tolika sloupci, kolik má testovaný obvod uzlů,
- b) do tabulky vpisujeme postupně řádky D-krychlí konstrukčních prvků testovaného obvodu, přičemž v souladu se schéma testovaného obvodu ztotožňujeme čísla vývodů konstrukčních prvků s čísly sloupců tabulky,
- c) ke každému sloupci tabulky singulárního pokrytí testovaného obvodu musí být přiřazena jedna výstupní špička konstrukčního prvku jako zdroje signálu v tomto uzlu a minimálně jedna vstupní špička následujícího konstrukčního prvku testovaného obvodu

VI. část sestavení kroku testu - D-šíření z místa poruchy na výstup testovaného obvodu

- a) sestavíme tabulku konstrukce kroku testu (**D-tabulku**). Tato tabulka musí mít tolik sloupců, kolik má testovaný obvod uzlů,
- b) do D-tabulky vepíšeme první řádek D-krychle poruchy, čímž jsme vnesli prvotní podmínky nastavení logických úrovní v uzlech testovaného obvodu pro demonstrování poruchy na výstupu obvodu. V některých případech může mít D-krychle poruchy pouze jeden řádek, v jiných případech jich může mít více v závislosti na konfiguraci konkrétního konstrukčního prvku. V každém případě je třeba prověřit samostatně všechny možné případy, i když některé nepovedou k výsledku,
- c) do D-tabulky vkládáme (deklarujeme) řádky D-pokrytí testovaného obvodu tak, aby k deklarované výstupní špičce D-pokrytí testovaného obvodu byly připojeny vhodné řádky D-pokrytí testovaného obvodu obsahující vstupní špičku připojenou k tomuto uzlu. Tento proces budeme dále nazývat D-přetínání. Kritériem výběru řádků D-pokrytí testovaného obvodu jsou vstupní špičky, které porovnáváme s deklarovanou hodnotou sloupce D-tabulky (uzlu testovaného obvodu). Postupnou aplikací D-přetínání je symbol D nebo \bar{D} šířen na výstup testovaného obvodu. Tato fáze je interpretací otevření cesty poruchy na výstup, tak jak byla popsána v bodě II. a) kapitoly 1.3.1.3,
- d) D-přetínání se uskutečňuje porovnáváním a vyhodnocováním hodnot jednotlivých řádků D-pokrytí testovaného obvodu, které mají v příslušných uzlech (ve stejných sloupcích) uvedeny vstupy nebo výstupy konstrukčních prvků testovaného obvodu. Protože se jedná o algoritmus, je třeba mechanicky projít všechny možnosti a dále vybrat aplikovatelné řádky D-pokrytí testovaného obvodu. Pokud D-algoritmus aplikuje diagnostik, snadno vybere pouze řádky připadající do úvahy,

- e) výběr aplikovatelných řádků D-pokrytí testovaného obvodu je dán možností současného výskytu více stavů vývodu konstrukčního prvku vyjádřených jeho D-krychlí. Pravidla výběru lze vyjádřit následující tabulkou **D-přetnutí**:

	0	1	X	D	\bar{D}	' '
0	0	Φ	0	ψ	ψ	0
1	Φ	1	1	ψ	ψ	1
X	0	1	X	D	\bar{D}	X
D	ψ	ψ	D	μ	λ	D
\bar{D}	ψ	ψ	\bar{D}	λ	μ	\bar{D}
' '	0	1	X	D	\bar{D}	' '

Tab. 1.3: výsledky D-přetnutí

- f) význam jednotlivých symbolů je následující:
- 0** = v uzlu testovaného obvodu musí být úroveň **log.0**,
 - 1** = v uzlu testovaného obvodu musí být úroveň **log.1**,
 - D** = tímto uzlem je přenášen projev poruchy na výstup,
 - \bar{D}** = tímto uzlem je přenášen inverzní projev poruchy na výstup,
 - Φ** = prázdný průnik, tento řádek nelze použít,
 - ψ** = nedefinovaný průnik, tento řádek nelze použít,
 - μ** = zůstane zachován smysl změn, symboly **D** a **\bar{D}** se nemění,
 - λ** = bude změněn smysl změn **\bar{D}** a **D**, tyto symboly budou zaměněny. Vyjadřuje se tím změna smyslu přechodu z **log.0** do **log.1** na z **log.1** do **log.0**. Tato změna má význam pouze v případě, kdy nás zajímá chybová úroveň na výstupu testovaného obvodu,
 - ' ' = mezer - prozatím nedeklarovaný uzel testovaného obvodu,

VII. část sestavení kroku testu - zpětné šíření ke vstupu, odvození kroku testu

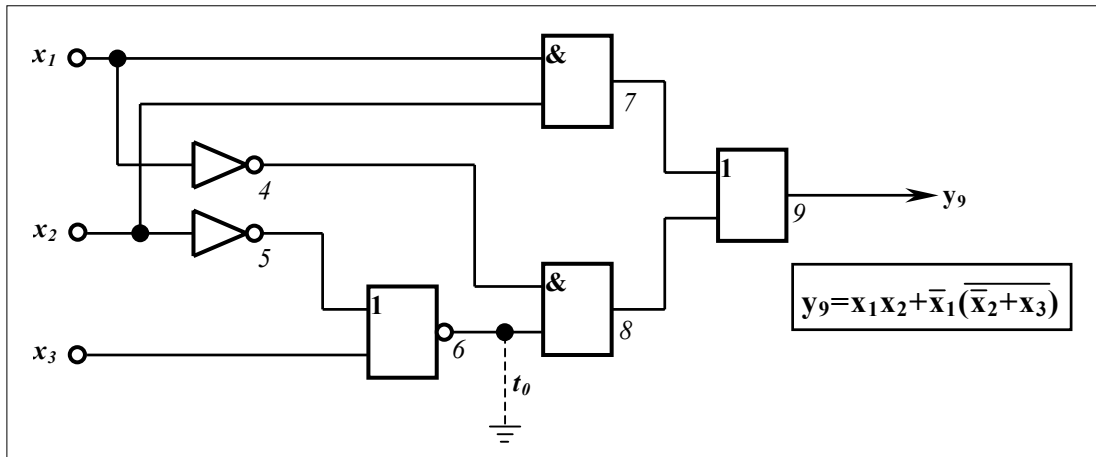
- a) pokud je v posledním sloupci D-tabulky deklarován symbol **\bar{D}** nebo **D**, nastává fáze zajištění podmínek zpětného šíření poruchy. Při zpětném šíření do D-šíření vstupují vhodné řádky tabulky singulárního pokrytí testovaného obvodu směrem od výstupu (uzlu s nejvyšším číslem) ke vstupním uzlům testovaného obvodu. V této fázi ověřujeme možnost dodržení podmínek pro otevření cesty projevu poruchy na výstup testovaného obvodu a zároveň nastavení podmínek pro demonstraci poruchy - nastavení podmínek pro nastavení inverzní hodnoty v místě poruchy (inverzní vůči projevu poruchy),
- b) kritériem D-protínání jsou výstupy v tabulce singulárního pokrytí testovaného obvodu, podle něhož se vybírají akceptovatelné řádky,
- c) postupně se ověřuje nastavení hodnot ve všech uzlech testovaného obvodu (ve všech sloupcích D-tabulky). Je nutno prověřit všechny vyhovující řádky tabulky singulárního pokrytí testovaného obvodu,

d) pokud tímto postupem dospějeme bez konfliktů až ke vstupům testovaného obvodu, je nalezen krok testu.

Postup získání kroku testu bude konkrétně vysvětlen a znázorněn v následujícím příkladu č. 1.6.

Příklad 1.6:

Nalezněte pomocí D-algoritmu krok testu pro poruchu t_0 v uzlu 6 obvodu na obrázku č. 1.9. Poruchu t_0 můžeme zobrazit jako zkrat uzlu 6 na zemní vodič.

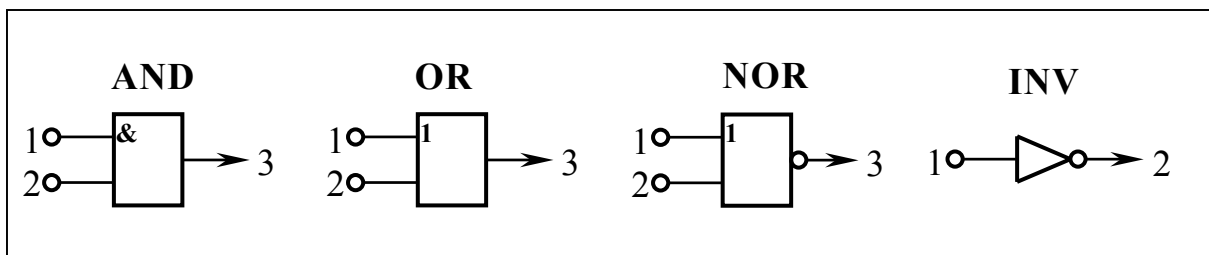


Obr.č.1.9: Schéma obvodu s poruchou t_0 v uzlu 6.

Postup:

V zájmu rychlejšího postupu a vzhledem k podobnosti operací budou při řešení příkladu některé fáze popsány společně.

I., II. - tabulky singulárního pokrytí a D-krychle konstrukčních prvků testovaného obvodu



Tabulky singulárního pokrytí konstrukčních prvků obvodu:

1	2	3
0	X	0
X	0	0
1	1	1

1	2	3
1	X	1
X	1	1
0	0	0

1	2	3
1	X	0
X	1	0
0	0	1

1	2
1	0
0	1

Tabulky D-krychlí konstrukčních prvků obvodu:

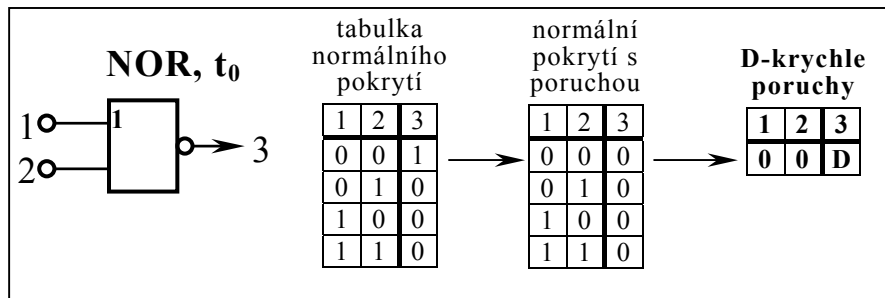
1	2	3
D	1	D
1	D	D

1	2	3
D	0	D
0	D	D

1	2	3
D	0	D
0	D	D

1	2
D	D

III. - konstrukce D-krychle poruchy



Pro poruchu t_0 má obvod typu NOR pouze jednu D-krychli poruchy, kterou označíme symbolem K_1 .

IV.a V. - singulární tabulka pokrytí a D-pokrytí testovaného obvodu

Pro snadnější orientaci budou jednotlivé řádky tabulky singulárního pokrytí a D-pokrytí testovaného obvodu označovány podle uzlu kam je připojen výstup konstrukčního prvku a číslem řádku tabulek singulárního pokrytí a D-krychle konstrukčního prvku.

Singulární pokrytí

	1	2	3	4	5	6	7	8	9
H4a	1			0					
H4b	0			1					
H5a		1			0				
H5b		0			1				
H6a			1		X	0			
H6b			X		1	0			
H6c			0		0	1			
H7a	0	X					0		
H7b	X	0					0		
H7c	1	1					1		
H8a				0		X		0	
H8b				X		0		0	
H8c				1		1		1	
H9a							1	X	1
H9b							X	1	1
H9c							0	0	0

D-pokrytí testovaného obvodu

	1	2	3	4	5	6	7	8	9	
H _D 4	D			\bar{D}						
H _D 5		D			\bar{D}					
H _D 6a			D		0	\bar{D}				
H _D 6b			0		D	\bar{D}				
H _D 7a	D	1						D		
H _D 7b	1	D						D		
H _D 8a				D		1		D		
H _D 8b				1		D		D		
H _D 9a								D	0	D
H _D 9b								0	D	D

VI. - D-tabulka a šíření z místa poruchy na výstup testovaného obvodu

Pro šíření poruchy podle D-krychle poruchy K_1 na výstup byly

vybrány do *D*-tabulky pouze zjevně použitelné řádky tabulky *D*-pokrytí testovaného obvodu. Algoritmus by zkoušel všechny řádky *D*-pokrytí testovaného obvodu a vyhodnocoval je podle tab. 1.3.

K_1	1	2	3	4	5	6	7	8	9
			0		0	D			
H_D8b				1		D		D	
protnutí			0	1	0	D		D	
H_D9b							0	D	D
protnutí			0	1	0	D	0	D	D

VII. - zpětné šíření ke vstupu, odvození kroku testu

V tomto okamžiku je určeno, jaké musí být logické hodnoty v některých bodech testovaného obvodu. V dalším postupu budeme používat tabulku singulárního pokrytí testovaného obvodu pro ověření podmínek zpětného šíření a nastavení vstupů hradla s poruchou do inverzní hodnoty.

V tomto příkladě jsou do *D*-protínání pro *D*-krychli poruchy K_1 vybírány řádky, které dávají použitelné výsledky. Algoritmus by zkoušel všechny řádky Singulárního pokrytí testovaného obvodu a vyhodnocoval je podle tab. 1.3.

K_1	1	2	3	4	5	6	7	8	9
			0	1	0	D	0	D	D
H7a	0	X					0		

v této fázi lze použít i řádek *H7b* - vrátíme se k němu po ukončení této varianty zpětného šíření

protnutí	0	X	0	1	0	D	0	D	D
H5a		1			0				
protnutí	0	1	0	1	0	0	D	D	D
H4b	0			1					
protnutí	0	1	0	1	0	0	D	D	D

Pro zachování pravidel použití *D*-algoritmu se vrátíme k ověření hodnoty v uzlu 7 a použitím řádku *H7b*.

K_1	1	2	3	4	5	6	7	8	9
			0	1	0	D	0	D	D
H7b	X	0					0		
protnutí	X	0	0	1	0	D	0	D	D
H5a		1			0				
protnutí	X	0	0	1	0	0	D	D	D

Protože ve sloupci 2 po aplikaci řádků *H7b* a *H5a* vzniká prázdný průnik, algoritmus ukončuje svou činnost - v tomto případě nelze

odvodit krok testu

Řešení:

Krok testu pro poruchu t_0 v uzlu 6 představuje vstupní kombinace:

$$x_1, x_2, x_3 = 0, 1, 0 \text{ (ekvivalent 2)}$$

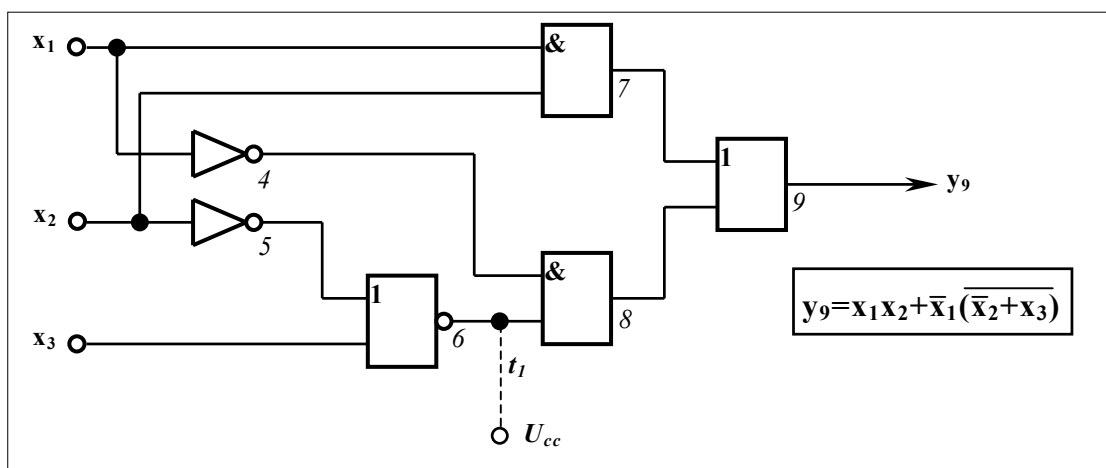
Za povšimnutí stojí dvě vlastnosti D-algoritmu. První podstatnou vlastností je to, že jak singulární pokrytí a D-krychle konstrukčních prvků, tak D-krychle poruchy jsou trvale platné. Druhou podstatnou vlastností D-algoritmu je skutečnost, že singulární pokrytí testovaného obvodu a D-pokrytí testovaného obvodu jsou pro konkrétní testovaný obvod konstantní a pro všechny možné poruchy testovaného obvodu neměnné. Tyto vlastnosti D-algoritmu jej činí velmi výkonným nástrojem pro automatizované sestavování kroků testů číslicových obvodů.

Pro ilustraci této vlastnosti D-algoritmu použijeme další příklady.

Poněkud složitější situaci, než předchozí příklad, představuje situace, kdy v uzlu 6 testovaného obvodu je porucha typu t_1 . V předchozím příkladu existují celkem tři krychle poruchy a D-algoritmus dopředu nedokáže určit, která z nich povede k cíli a která nikoliv. Řešení je ukázáno v příkladě č. 1.7.

Příklad 1.7:

Nalezněte pomocí D-algoritmu krok testu pro poruchu t_1 v uzlu 6 obvodu na obrázku č. 1.10.



Obr.č.1.10: Schéma obvodu s poruchou t_1 v uzlu 6.

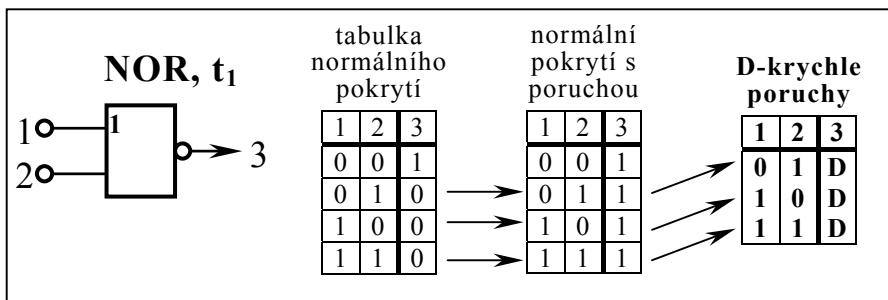
Porucha t_1 obvodu je zobrazena jako připojení uzlu 6 na napájecí napětí obvodu. Podstatné pro tuto úlohu je, že obvod 8 pracuje s úrovní log.1 na vstupu 6.

Postup:

I., II. - tabulky singulárního pokrytí a D- krychle konstrukčních prvků testovaného obvodu

Obě tabulky jsou shodné s tabulkami uvedenými v příkladě 1.6.

III. - konstrukce D-krychle poruchy



Porucha t_1 má celkem tři D-krychle, které označíme K_2 , K_3 a K_4 . V tomto příkladě musíme vyzkoušet použitelnost všech tří možností, protože dopředu nemůžeme říci, která D-krychle poruchy povede k nalezení kroku testu. Často se u redundantních a neminimalizovaných obvodů (což je i náš příklad) stává, že i různé D-krychle poruch ukazují na tutéž vstupní kombinaci.

IV., V. - singulární tabulka pokrytí a D-pokrytí testovaného obvodu

Singulární tabulka pokrytí a tabulka D-pokrytí testovaného obvodu jsou shodné s tabulkami uvedenými v příkladě 1.6.

VI. - D-tabulka a šíření z místa poruchy na výstup testovaného obvodu

Pro šíření poruchy podle D-krychle poruchy K_2 na výstup byly vybrány do D-tabulky pouze zjevně použitelné řádky tabulky D-pokrytí testovaného obvodu. Algoritmus by zkoušel všechny řádky D-pokrytí testovaného obvodu a vyhodnocoval je podle tab. 1.3.

K_2	1	2	3	4	5	6	7	8	9
			0		1	D			
$H_D 8b$				1		D		D	
protnutí			0	1	1	D		D	
$H_D 9b$							0	D	D
protnutí			0	1	1	D	0	D	D

Pro šíření poruchy podle D-krychle poruchy K_3 na výstup byly vybrány do D-tabulky pouze zjevně použitelné řádky tabulky D-pokrytí testovaného obvodu. Algoritmus by zkoušel všechny řádky D-pokrytí testovaného obvodu a vyhodnocoval je podle tab. 1.3.

K_3	1	2	3	4	5	6	7	8	9
			1		0	D			
$H_D 8b$				1		D		D	
protnutí			1	1	0	D		D	
$H_D 9b$							0	D	D
protnutí			1	1	0	D	0	D	D

Pro šíření poruchy podle D-krychle poruchy K_4 na výstup byly

vybrány do *D*-tabulky pouze zjevně použitelné řádky tabulky *D*-pokrytí testovaného obvodu. Algoritmus by zkoušel všechny řádky *D*-pokrytí testovaného obvodu a vyhodnocoval je podle tab. 1.3.

K_4	1	2	3	4	5	6	7	8	9
			1		1	D			
H_{p8b}				1		D		D	
protnutí			1	1	1	D		D	
H_{p9b}							0	D	D
protnutí			1	1	1	D	0	D	D

VII. - zpětné šíření ke vstupu, odvození kroku testu

V tomto okamžiku je určeno, jaké musí být logické hodnoty v některých bodech testovaného obvodu. V dalším postupu budeme používat tabulku singulárního pokrytí testovaného obvodu pro ověření podmínek zpětného šíření a nastavení vstupů hradla s poruchou do inverzní hodnoty.

Pro sestavení zpětného šíření a nastavení vstupů hradla s poruchou do inverzní hodnoty podle *D*-krychle poruchy K_2 byly do *D*-tabulky vybírány pouze zjevně použitelné řádky tabulky Singulárního pokrytí testovaného obvodu, které dávají použitelné výsledky. Algoritmus musí zkoušet všechny řádky Singulárního pokrytí testovaného obvodu a vyhodnocoval je podle tab. 1.3.

K_2	1	2	3	4	5	6	7	8	9
			0	1	1	D	0	D	D
$H7a$	0	X					0		

v této fázi lze použít i řádek $H7b$ - je zjevné, že při jeho použití dostaneme stejný krok testu, protože ve sloupci 2 by od řádku $H7b$ byl znak 0 a řádek $H5b$ vyžaduje v tomto sloupci znak 0, viz. níže

protnutí	0	X	0	1	1	D	0	D	D
$H5b$		0			1				
protnutí	0	0	0	1	1	D	0	D	D
$H4b$	0			1					
protnutí	0	0	0	1	1	D	0	D	D

Pro sestavení zpětného šíření podle *D*-krychle poruchy K_3 byly do *D*-tabulky vybírány pouze zjevně použitelné řádky tabulky Singulárního pokrytí testovaného obvodu.

K_3	1	2	3	4	5	6	7	8	9
			1	1	0	D	0	D	D
$H7a$	0	X					0		

v této fázi lze použít i řádek $H7b$ - pro tuto krychli poruchy bychom se však v další fázi dostali do situace, kdy ve sloupci 2 dostáváme prázdný průnik,

protože ve sloupci 2 by od řádku **H7b** byl znak 0 a řádek **H5a** vyžaduje v tomto sloupci znak 1

protnutí	0	X	1	1	0	D	0	D	D
H5a		1			0				
protnutí	0	1	1	1	0	D	0	D	D
H4b	0			1					
protnutí	0	1	1	1	0	D	0	D	D

Pro sestavení zpětného šíření podle D-krychle poruchy K_4 byly do D-tabulky vybírány pouze zjevně použitelné řádky tabulky Singulárního pokrytí testovaného obvodu.

K_4	1	2	3	4	5	6	7	8	9
			1	1	1	D	0	D	D
H7a	0	X					0		

v této fázi lze použít i řádek **H7b** - obdobně jako u krychle poruchy K_2 je zjevné, že při jeho použití dostaneme stejný krok testu

protnutí	0	X	1	1	1	D	0	D	D
H5b		0			1				
protnutí	0	0	1	1	1	D	0	D	D
H4b	0			1					
protnutí	0	0	1	1	1	D	0	D	D

Řešení:

Krok testu pro poruchu t_1 v uzlu 6 testovaného obvodu tvoří následující vstupní kombinace (testovací vektory):

$$x_1, x_2, x_3 = 0, 0, 0 \text{ (ekvivalent 0),}$$

$$x_1, x_2, x_3 = 0, 0, 1 \text{ (ekvivalent 1),}$$

$$x_1, x_2, x_3 = 0, 1, 1 \text{ (ekvivalent 3).}$$

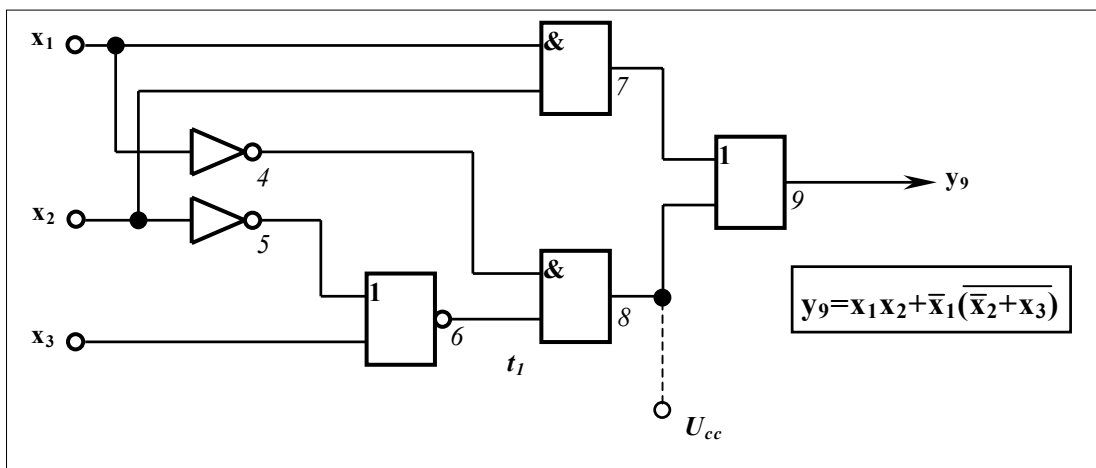
Předchozí dva příklady představovaly případy, kdy se během hledání kroku testu nevyskytly zásadní problémy při hledání výsledků protínání řádků v D-tabulce.

D-algoritmus představuje vhodný nástroj pro automatizované vyhledávání kroků testu. Při jeho ručním použití se často vyskytuje množství variant zejména při zpětném šíření a při aplikaci konstrukčních prvků s více vstupy, kdy nastaveným podmínkám vyhovuje více řádků tabulky singulárního pokrytí testovaného obvodu, ale ne všechny vedou k nalezení kroku testu.

Pro přiblížení možných nesrovnalostí při hledání kroku testu je v následujícím textu uveden ještě další příklad.

Příklad 1.8:

Nalezněte pomocí D-algoritmu krok testu pro poruchy t_1 v uzlu 8 obvodu na obrázku č. 1.11.



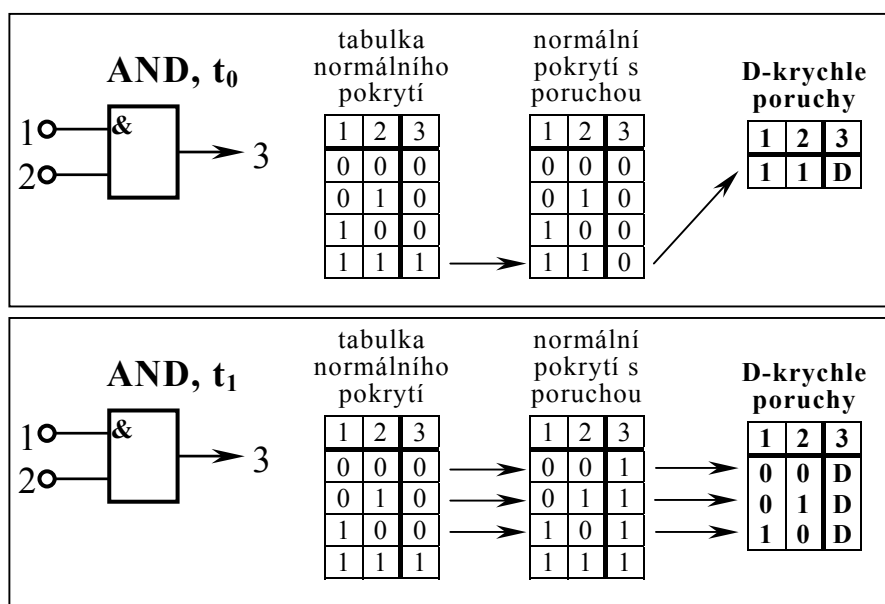
Obr.č.1.11: Schéma obvodu s poruchou t_1 v uzlu 8.

Postup:

I., II. - tabulky singulárního pokrytí a D- krychle konstrukčních prvků testovaného obvodu

Obě tabulky jsou shodné s tabulkami uvedenými v příkladě 1.6.

III. - konstrukce D-krychle poruchy obvodu typu AND



Pro porucha t_0 je D-krychle poruchy $K_1 - 11D$. Porucha t_1 obvodu typu AND je popsána třemi D-krychlemi poruchy: $K_2 - 00D$, $K_3 - 01D$ a $K_4 - 10D$. Nutno vyzkoušet použitelnost všech tří možností. Předem nelze usuzovat na použitelnost jednotlivých D-krychlí poruchy. U redundantních a neminimalizovaných obvodů (což je i náš příklad) stává, že i různé D-krychle poruch ukazují na tutéž vstupní kombinaci.

IV., V. - singulární tabulka pokrytí a D-pokrytí testovaného obvodu

Singulární tabulka pokrytí a tabulka D-pokrytí testovaného obvodu jsou shodné s tabulkami uvedenými v příkladě 1.6. Pro přehlednost jsou ve zmenšené podobě uvedeny níže.

Singulární pokrytí

	1	2	3	4	5	6	7	8	9
H4a	1			0					
H4b	0			1					
H5a		1			0				
H5b		0			1				
H6a			1		X	0			
H6b			X		1	0			
H6c			0		0	1			
H7a	0	X					0		
H7b	X	0					0		
H7c	1	1					1		
H8a				0		X		0	
H8b				X		0		0	
H8c				1		1		1	
H9a							1	X	1
H9b							X	1	1
H9c							0	0	0

D-pokrytí

	1	2	3	4	5	6	7	8	9
H _p 4	D			\bar{D}					
H _p 5		D			\bar{D}				
H _p 6a			D		0	\bar{D}			
H _p 6b			0		D	\bar{D}			
H _p 7a	D	1					D		
H _p 7b	1	D					D		
H _p 8a				D		1		D	
H _p 8b				1		D		D	
H _p 9a							D	0	D
H _p 9b							0	D	D

D-přetnutí

	0	1	X	D	\bar{D}	
0	0	Φ	0	ψ	ψ	0
1	Φ	1	1	ψ	ψ	1
X	0	1	X	D	\bar{D}	X
D	ψ	ψ	D	μ	λ	D
\bar{D}	ψ	ψ	\bar{D}	λ	μ	\bar{D}
0	1	X	D	\bar{D}		

VI. a VII. - D-tabulka a šíření z místa poruchy na výstup testovaného obvodu a zpětné šíření ke vstupu, odvození kroku testu

Pro zjednodušení popisu nebude dále uváděn kompletní komentář. Pouze bude upozorněno na případné zvláštnosti postupu.

K_1 v1	1	2	3	4	5	6	7	8	9	
1				0	0		D			
2	H _p 9b						0	D	D	
3	protnutí			0	0	0	D	D		cesta na výstup otevřena
4	H7a	0	X				0			lze použít i H7b
5	protnutí	0	X	0	0	0	D	D		
6	H6a			1	X	0				lze použít i H6b
7	protnutí	0	X	1	0	X	0	0	D	D
8	H5a		1		0					lze použít i H5b
9	protnutí	0	1	1	0	0	0	0	D	D
10	H4a	1			0					
11	protnutí	Φ	1	1	0	0	0	0	D	D

Po nastalém konfliktu je nutno se vrátit k nejbližší možnosti volby v singulárním pokrytí testovaného obvodu. Vrátime se k řádce 8 a použijeme H5b

K_1 v2	1	2	3	4	5	6	7	8	9	
1				0	0		D			
2	H _p 9b						0	D	D	
3	protnutí			0	0	0	D	D		cesta na výstup otevřena
4	H7a	0	X				0			lze použít i H7b
5	protnutí	0	X	0	0	0	D	D		
6	H6a			1	X	0				lze použít i H6b
7	protnutí	0	X	1	0	X	0	0	D	D
8	H5b		0		1					
9	protnutí	0	0	1	0	1	0	0	D	D
10	H4a	1			0					
11	protnutí	Φ	1	1	0	0	0	0	D	D

Tento konflikt bylo možno předpokládat. Pokud si prohlédneme řádky 8,9,10 a 11, je zřejmé že by opět v uzlu 1 došlo ke konfliktu, protože řádek H5b nemění podmínky v uzlu 1 testovaného obvodu. Vrátime se tudíž k řádce 6 a použijeme H6b

	K_1 v3	1	2	3	4	5	6	7	8	9	
1					0	0			D		
2	H _p 9b							0	D	D	
3	protnutí				0	0	0		D	D	cesta na výstup otevřena
4	H7a	0	X				0				lze použít i H7b
5	protnutí	0	X		0	0	0		D	D	
6	H6b			X		1	0				
7	protnutí	0	X	X	0	1	0	0	D	D	
8	H5b		0			1					
9	protnutí	0	0	X	0	1	0	0	D	D	
10	H4a	1			0						
11	protnutí	0	0	X	0	0	0	0	D	D	v uzlu 1 konflikt

Opět konflikt, je nutno se vrátit k řádce 4 a použít H7b

	K_1 v4	1	2	3	4	5	6	7	8	9	
1					0	0			D		
2	H _p 9b							0	D	D	
3	protnutí				0	0	0		D	D	cesta na výstup otevřena
4	H7b	X	0					0			
5	protnutí	X	0		0	0	0		D	D	
6	H6a			1		X	0				lze použít i H6b
7	protnutí	X	0	1	0	X	0	0	D	D	
8	H5b		0			1					lze použít i H5a
9	protnutí	X	0	1	0	1	0	0	D	D	
10	H4a	1			0						
11	protnutí	1	0	1	0	1	0	0	D	D	testovací vektor

Nutno prověřit všechny možnosti, nutno se vrátit k řádce 8 a použít H5b

	K_1 v5	1	2	3	4	5	6	7	8	9	
1					0	0			D		
2	H _p 9b							0	D	D	
3	protnutí				0	0	0		D	D	cesta na výstup otevřena
4	H7b	X	0					0			
5	protnutí	X	0		0	0	0		D	D	
6	H6a			1		X	0				lze použít i H6b
7	protnutí	X	0	1	0	X	0	0	D	D	
8	H5a		1		0						
9	protnutí	X	0	1	0	0	0	0	D	D	konflikt v uzlu 2

Zde varianta 5 předčasně končí. Zbývá prověřit řádek 6 a použít H6b

	K_1 v6	1	2	3	4	5	6	7	8	9	
1					0	0			D		
2	H _p 9b							0	D	D	
3	protnutí				0	0	0		D	D	cesta na výstup otevřena
4	H7b	X	0					0			
5	protnutí	X	0		0	0	0		D	D	
6	H6b			X		1	0				
7	protnutí	X	0	X	0	1	0	0	D	D	
8	H5b		0			1					
9	protnutí	X	0	X	0	1	0	0	D	D	
10	H4a	1			0						
11	protnutí	1	0	X	0	1	0	0	D	D	2 testovací vektory

Pro D-krychli poruchy K_1 dostáváme dva testovací vektory. Nyní prověříme použitelnost D-krychle poruchy K_2 .

	K_2 v1	1	2	3	4	5	6	7	8	9	
1					0		1		D		
2	H_{D9b}							0	D	D	
3	protnutí				0		1	0	D	D	<i>cesta na výstup otevřena</i>
4	$H7a$	0	X					0			<i>lze použít i $H7b$</i>
5	protnutí	0	X		0		1	0	D	D	
6	$H6c$			0		0	1				
7	protnutí	0	X	0	0	0	1	0	D	D	
8	$H5a$		1			0					
9	protnutí	0	1	0	0	0	1	0	D	D	
10	$H4a$	1			0						
11	protnutí	0	1	0	0	0	0	0	D	D	<i>v uzlu 1 konflikt</i>

Po nastalém konfliktu je nutno se vrátit k nejbližšímu možnosti volby v singulárním pokrytí testovaného obvodu. Vrátime s k řádce 4 a použijeme $H7b$

	K_2 v2	1	2	3	4	5	6	7	8	9	
1					0		1		D		
2	H_{D9b}							0	D	D	
3	protnutí				0		1	0	D	D	<i>cesta na výstup otevřena</i>
4	$H7b$	X	0					0			
5	protnutí	X	0		0		1	0	D	D	
6	$H6c$			0		0	1				
7	protnutí	X	0	0	0	0	1	0	D	D	
8	$H5a$		1			0					
9	protnutí	0	0	0	0	0	1	0	D	D	<i>v uzlu 2 konflikt</i>

Pro D -krychli poruchy K_2 nedostáváme žádné řešení. Budeme pokračovat prověřováním použitelnosti D -krychle poruchy K_3 .

	K_3 v1	1	2	3	4	5	6	7	8	9	
1					1		0		D		
2	H_{D9b}							0	D	D	
3	protnutí				1		0	0	D	D	<i>cesta na výstup otevřena</i>
4	$H7a$	0	X					0			<i>lze použít i $H7b$</i>
5	protnutí	0	X		1		0	0	D	D	
6	$H6a$			1		X	0				<i>lze použít i $H6b$</i>
7	protnutí	0	X	1	1	X	0	0	D	D	
8	$H5a$		1			0					<i>lze použít i $H5b$</i>
9	protnutí	0	1	1	1	0	0	0	D	D	
10	$H4b$	0			1						
11	protnutí	0	1	1	1	0	0	0	D	D	<i>testovací vektor</i>

Nutno prověřit všechny možnosti, nutno se vrátit k řádce 8 a použít $H5b$

	K_3 v2	1	2	3	4	5	6	7	8	9	
1					1		0		D		
2	H_{D9b}							0	D	D	
3	protnutí				1		0	0	D	D	<i>cesta na výstup otevřena</i>
4	$H7a$	0	X					0			<i>lze použít i $H7b$</i>
5	protnutí	0	X		1		0	0	D	D	
6	$H6a$			1		X	0				<i>lze použít i $H6b$</i>
7	protnutí	0	X	1	1	X	0	0	D	D	
8	$H5b$		0			1					
9	protnutí	0	0	1	1	1	0	0	D	D	
10	$H4b$	0			1						
11	protnutí	0	0	1	1	1	0	0	D	D	<i>testovací vektor</i>

Nutno prověřit všechny možnosti, nutno se vrátit k řádce 6 a použít **H6b**

	K_3 v3	1	2	3	4	5	6	7	8	9	
1					1	0			D		
2	H_p9b						0		D	D	
3	protnutí				1	0	0		D	D	cesta na výstup otevřena
4	H7a	0	X				0				lze použít i H7b
5	protnutí	0	X		1	0	0		D	D	
6	H6b			X		1	0				
7	protnutí	0	X	X	1	1	0		D	D	
8	H5b		0			1					
9	protnutí	0	0	X	1	1	0		D	D	
10	H4b	0			1						
11	protnutí	0	0	X	1	1	0		D	D	dva testovací vektory

Nutno prověřit všechny možnosti, nutno se vrátit k řádce 4 a použít **H7b**

	K_3 v4	1	2	3	4	5	6	7	8	9	
1					1	0			D		
2	H_p9b						0		D	D	
3	protnutí				1	0	0		D	D	cesta na výstup otevřena
4	H7b	X	0				0				
5	protnutí	X	0		1	0	0		D	D	
6	H6a			1		X	0				lze použít i H6b
7	protnutí	X	0	1	1	X	0		D	D	
8	H5a		1			0					lze použít i H5b
9	protnutí	0	0	X	1	1	0		D	D	v uzlu 2 konflikt

Zde varianta 4 předčasně končí. Nutno se vrátit k řádce 8 a použít **H5b**

	K_3 v5	1	2	3	4	5	6	7	8	9	
1					1	0			D		
2	H_p9b						0		D	D	
3	protnutí				1	0	0		D	D	cesta na výstup otevřena
4	H7b	X	0				0				
5	protnutí	X	0		1	0	0		D	D	
6	H6a			1		X	0				lze použít i H6b
7	protnutí	X	0	1	1	X	0		D	D	
8	H5b		0			1					
9	protnutí	0	0	1	1	1	0		D	D	
10	H4b	0			1						
11	protnutí	0	0	1	1	1	0		D	D	testovací vektor

Zbývá prověřit poslední variantu v řádce 6 použít **H6b**

	K_3 v6	1	2	3	4	5	6	7	8	9	
1					1	0			D		
2	H_p9b						0		D	D	
3	protnutí				1	0	0		D	D	cesta na výstup otevřena
4	H7b	X	0				0				
5	protnutí	X	0		1	0	0		D	D	
6	H6b			X		1	0				
7	protnutí	X	0	X	1	1	0		D	D	
8	H5b		0			1					
9	protnutí	0	0	X	1	1	0		D	D	
10	H4b	0			1						
11	protnutí	0	0	X	1	1	0		D	D	dva testovací vektory

*Tímto okamžikem vyčerpány všechny možnosti zpětného šíření a zbývá definovat kroky testu pro poruchu t_1 v uzlu **8** testovaného obvodu.*

Odvození kroku testu spočívá v nalezení všech vstupních kombinací vyhovující získaným výsledkům. Jedná se o následující kombinace: $K_1v4 - 101$, $K_1v6 - 10X$ (100 a 100), $K_3v1 - 011$, $K_3v2 - 001$, $K_3v3 - 00X$ (000 a 001), $K_3v5 - 001$ a $K_3v6 - 00X$ (000 a 001). Těmto podmínkám vyhovují následující vstupní vektory: 000 , 001 , 011 , 100 a 101 .

Řešení:

*Krokem testu pro poruchu t_1 v uzlu **8** jsou následující vstupní vektory:*

$x_1, x_2, x_3 = 0, 0, 0$ (ekvivalent 0),

$x_1, x_2, x_3 = 0, 0, 1$ (ekvivalent 1),

$x_1, x_2, x_3 = 0, 1, 1$ (ekvivalent 3),

$x_1, x_2, x_3 = 1, 0, 0$ (ekvivalent 4),

$x_1, x_2, x_3 = 1, 0, 1$ (ekvivalent 5),

Nalezení kroku testu pro poruchu t_0 v uzlu **8** testovaného obvodu je ponechána na čtenáři. Postup je zcela obdobný a všechny zvláštnosti D-algoritmu již byly v příkladech popsány.

1.3.2 Doslov k sestavení kroku testu

Na předložených příkladech je doloženo, že všechny metody sestavení kroku testu vedou ke shodným výsledkům. Výběr metody závisí na účelu a prostředcích, které má diagnostik k dispozici.

V současnosti je hledání a sestavování testů přenecháno návrhovým systémům elektronických obvodů, které pracují na principech uvedených v předchozí kapitole. V systémech se zvýšenou spolehlivostí a v systémech odolných poruchám se předpokládá použití těchto návrhových systémů jako standard.

Řada aplikací pro státní správu považuje automatizované sestavení testovacích prostředků a jejich předání uživateli nebo servisním organizacím za zcela samozřejmý předpoklad realizace obchodního vztahu. V zadávacích dokumentacích akvizic řady elektronických systémů je návrh schváleným, návrhovým systémem samozřejmým předpokladem úspěchu ve výběrovém řízení.

Soudobé návrhové systémy elektronických zařízení kromě vlastního návrhu schéma zařízení zajišťují návrh kladu součástek, návrh plošného spoje, podklady nebo přímo řídicí programy pro linku na výrobu a osazování desek plošných spojů, energetickou a teplotní analýzu, podmínky elektromagnetické kompatibility a pochopitelně testovací rutiny pro ověření průceschopnosti dodaného zařízení.

1.4 Sestavení testů kombinačního obvodu

V dalším textu této kapitoly budeme hovořit pouze o číslicových kombinačních obvodech, a proto již tento fakt nebudeme v textu dále zdůrazňovat.

1.4.1 Sestavení detekčního testu

Pro zjednodušení úlohy budeme předpokládat vznik pouze jedné poruchy typu t_0 nebo t_1 v testovaném obvodu. Pro sestavení testu obvodu se používají diagnostické vektory všech možných poruch testovaného obvodu. V našem případě tedy v každém uzlu obvodu mohou vzniknout dvě poruchy. Každá porucha nějakým způsobem modifikuje činnost obvodu, dostáváme tak realizaci obvodu s konkrétní poruchou. Tuto realizaci označíme jako M_i realizaci obvodu (M_i je zkratkou termínu **MACHINE i**). Pokud uvažujeme bezchybný obvod jako jednu z realizací obvodu označenou jako M_0 a obvod má U uzlů, dostáváme celkem $2 \cdot U + 1$ možných variant realizace obvodu.

Výsledkem detekčního testu je informace, zda testovaný obvod je totožný s realizací M_0 či nikoliv. Postup sestavení detekčního testu je následující:

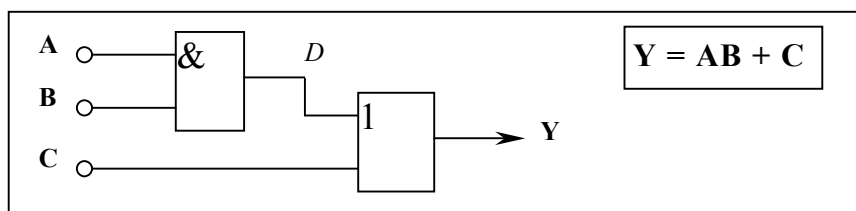
- I. nalezení testovacích vektorů (kroků testu) všech možných poruch,
- II. nalezení všech poruch pokrytých (detekovatelných) jedním konkrétním vektorem (jeden testovací vektor zpravidla detekuje více poruch, což je dáno podstatou činnosti číslicových obvodů),
- III. opakování bodu 2) pro všechny testovací vektory,
- IV. minimalizace množiny testovacích vektorů. Tu lze provádět různými způsoby. Na základě zkušenosti je doporučován následující postup:
 - i) první vektory vybíráme metodou půlení intervalu - počet pokrytých a nepokrytých poruch zvoleným testovacím vektorem má být přibližně stejný
 - ii) pokud je tato možnost volby vyčerpaná, vybíráme další testovací vektory, které pokrývají doposud neprověřované poruchy testovaného obvodu
 - iii) z případného většího počtu testovacích vektorů pokrývajících tutéž poruchu vybereme vždy pouze jeden
 - iv) postup opakujeme dokud nejsou pokryty všechny poruchy testovaného obvodu.

Postup sestavení detekčního testu může být pro složitější obvody zdouhavý, ale hlavně nepřehledný. Z tohoto důvodu bude demonstrován v následujícím příkladě na jednoduchém kombinačním obvodu.

Příklad 1.9:

Sestavte triviální test obvodu na obrázku č. 1.12 a sestavte jeho

detekční test.



Obr.č.1.12: Schéma obvodu pro sestavení detekčního testu

Postup:

A. Výsledek kroků I., II. a III. postupu vytvoření detekčního testu.

test	A	B	C	Y	E_A	E_B	E_C	E_D	E_Y
T ₀	0	0	0	0			t_1	t_1	t_1
T ₁	0	0	1	1			t_0		t_0
T ₂	0	1	0	0	t_1		t_1	t_1	t_1
T ₃	0	1	1	1			t_0		t_0
T ₄	1	0	0	0		t_1	t_1	t_1	t_1
T ₅	1	0	1	1			t_0		t_0
T ₆	1	1	0	1	t_0	t_0		t_0	t_0
T ₇	1	1	1	1					t_0

Vyplněná políčka sloupců označených E_I indikují změnu (inverzi) hodnoty na výstupu obvodu pro poruchu t_0 nebo t_1 v uzlu I. Nevyplněná políčka znamenají, že příslušný testovací vektor poruchu v uzlu I nedetekuje.

B. výběr prvního vektoru půlením intervalu - viz. bod IV.i) - pro poruchy typu t_1 . Testovacím vektorem byla zvolena vstupní kombinace T₂, která pokrývá 4 poruchy z 10 možných - optimální hodnota by měla být rovna 5. Pro poruchy typu t_1 lze vybrat jako první také vektor T₄, případně lze výběr prvního vektoru provést podle poruch typu t_0 - v tomto případě by se jednalo o vektor T₆,

test	A	B	C	Y	E_A	E_B	E_C	E_D	E_Y
T ₀	0	0	0	0			t_1	t_1	t_1
T ₁	0	0	1	1			t_0		t_0
T ₂	0	1	0	0	t_1		t_1	t_1	t_1
T ₃	0	1	1	1			t_0		t_0
T ₄	1	0	0	0		t_1	t_1	t_1	t_1
T ₅	1	0	1	1			t_0		t_0
T ₆	1	1	0	1	t_0	t_0		t_0	t_0
T ₇	1	1	1	1					t_0

C. výběr druhého vektoru podle bodu IV.i) pro doposud nepokrytou poruchu t_1 v uzlu **B**. Testovacím vektorem je vstupní kombinace T_4 . Nyní jsou pokryty všechny poruchy typu t_1 ,

test	A	B	C	Y	E_A	E_B	E_C	E_D	E_Y
T_0	0	0	0	0			t_1	t_1	t_1
T_1	0	0	1	1			t_0		t_0
T_2	0	1	0	0	t_1		t_1	t_1	t_1
T_3	0	1	1	1			t_0		t_0
T_4	1	0	0	0		t_1	t_1	t_1	t_1
T_5	1	0	1	1			t_0		t_0
T_6	1	1	0	1	t_0	t_0		t_0	t_0
T_7	1	1	1	1					t_0

D. výběr třetího vektoru podle bodu IV.i) pro poruchy t_0 (4 pokryté poruchy z 10 možných). Testovacím vektorem je kombinace T_6 ,

test	A	B	C	Y	E_A	E_B	E_C	E_D	E_Y
T_0	0	0	0	0			t_1	t_1	t_1
T_1	0	0	1	1			t_0		t_0
T_2	0	1	0	0	t_1		t_1	t_1	t_1
T_3	0	1	1	1			t_0		t_0
T_4	1	0	0	0		t_1	t_1	t_1	t_1
T_5	1	0	1	1			t_0		t_0
T_6	1	1	0	1	t_0	t_0		t_0	t_0
T_7	1	1	1	1					t_0

E. výběr čtvrtého testovacího vektoru podle bodu IV.ii) výběrem vstupní kombinace pro doposud nepokrytou poruchu typu t_0 v uzlu **C** (zcela rovnocennou možností poskytují také vstupní kombinace T_1 nebo T_3).

test	A	B	C	Y	E_A	E_B	E_C	E_D	E_Y
T_0	0	0	0	0			t_1	t_1	t_1
T_1	0	0	1	1			t_0		t_0
T_2	0	1	0	0	t_1		t_1	t_1	t_1
T_3	0	1	1	1			t_0		t_0
T_4	1	0	0	0		t_1	t_1	t_1	t_1
T_5	1	0	1	1			t_0		t_0
T_6	1	1	0	1	t_0	t_0		t_0	t_0
T_7	1	1	1	1					t_0

Řešení:

Úplný detekční test obvodu je tvořen vektory T_2 , T_4 , T_5 a T_6 .

1.4.2 Sestavení lokalizačního testu

Jak bylo konstatováno v předchozí kapitole, každá porucha nějakým způsobem modifikuje činnost obvodu, dostáváme tak realizaci obvodu s konkrétní poruchou. Tuto realizaci označíme jako \mathbf{M}_i realizaci obvodu (\mathbf{M}_i je zkratkou termínu **MACHINE i**), kde index i znamená pořadové číslo možné poruchy testovaného obvodu. Bezchybný obvod označujeme jako jednu z realizací obvodu symbolem \mathbf{M}_0 .

Pro sestavení lokalizačního testu lze použít stejné tabulky jaká je využívána pro sestavení detekčního testu obvodu. Tato tabulka však musí vycházet ze strukturního testu testovaného obvodu. Pro snadnější orientaci v tabulce je vhodné označit jednotlivé pokryté (detekovatelné) realizace obvodu, v souladu s pořadovým číslem poruchy, jako \mathbf{M}_i .

Sestavení lokalizačního testu spočívá ve výběru jednotlivých testovacích vektorů, které podle diagnostického příznaku rozdělují množinu možných realizací testovaného obvodu na pokryté (testované) a nepokryté (netestované). Podle diagnostického příznaku (porovnání hodnoty konkrétního výstupu s hodnotou výstupu obvodu bez poruchy, tedy realizace \mathbf{M}_0) se vybírají další realizace testovaného obvodu, které připadají do úvahy při dalším kroku lokalizačního testu. Konstrukce lokalizačního testu končí, pokud pro zbývající aktuální realizace obvodu nelze najít testovací vektor, který by tuto skupinu atomizoval.

Diagnostický příznak nabývá hodnoty "souhlasí", ve zkratce označované symbolem "+" nebo "nesouhlasí", ve zkratce označované symbolem "-". Hodnota "+" znamená, že testovacím vektorem prověřované realizace testovaného obvodu vykazují v tomto kroku lokalizačního testu správnou funkci, tedy funkci shodnou s realizací \mathbf{M}_0 . Naopak hodnota "-" znamená, že testovacím vektorem prověřované realizace testovaného obvodu vykazují v tomto kroku lokalizačního testu chybnou funkci, tedy funkci inverzní vůči realizaci \mathbf{M}_0 . Na této informaci je založen výběr dalšího testovacího vektoru lokalizačního testu. Pokud diagnostický příznak nabývá hodnoty "+", znamená to, že reálný testovaný obvod nemůže být shodný s některou z momentálně detekovatelných realizací testovaného obvodu. Pokud diagnostický příznak nabývá hodnoty "-", znamená to, že reálný testovaný obvod může být shodný s některou z momentálně detekovatelných realizací testovaného obvodu. Takto se celková množina realizací testovaného obvodu rozpadá na dílčí skupiny, nad nimiž můžeme výše popsany postup opakovat do té doby, pokud máme k dispozici vhodné testovací vektory dovolující další atomizaci dílčích skupin.

Teoreticky jsou možné dvě krajní varianty topologie lokalizačního testu. Pokud volíme testovací vektory tak, aby skupiny detekovatelných a

nedetekovatelných realizací testovacího obvodu byly stejně objemné, dostáváme symetrický lokalizační test - tento má optimální délku (stejný počet kroků pro jakoukoliv realizaci testovacího obvodu). Pokud volíme testovací vektory, které dílčí skupinu atomizují tak, že v jedné dílčí skupině je jedna realizace testovacího obvodu a v druhé dílčí skupině zbývající realizace, pak dostáváme asymetrický lokalizační test, který pro některou realizaci testovacího obvodu má počet kroků shodný s počtem realizací testovacího obvodu a pro jinou realizaci testovacího obvodu pouze jeden krok. Protože pravděpodobnost výskytu poruch v běžných číslicových obvodech je pro všechny konstrukční prvky stejná, je v praxi volena pokud možno symetrická forma lokalizačních testů.

Z podstaty fungování kombinačních obvodů vyplývá, že diagnostické příznaky některých realizací testovaného obvodu mohou být totožné, a tudíž je lokalizačním testem nelze jednoznačně rozlišit. Pro jednoznačnou lokalizaci poruchy je nezbytné provést měření logických hodnot signálů uvnitř reálného obvodu.

Postup sestavení lokalizačního testu:

- I. nalezení testovacích vektorů všech možných poruch,
- II. nalezení všech poruch pokrytých (prověřovaných) jedním konkrétním vektorem poruchy,
- III. opakování bodu II. pro všechny vektory poruch,
- IV. přiřazení realizací poruch obvodu detekovaným poruchám,
- V. výběr testovacího vektoru metodou půlení intervalu (počet detekovatelných a nedetekovatelných realizací testovacího obvodu zvoleným vektorem má být pokud možno přibližně stejný). Dostáváme tak skupinu pokrytých realizací testovaného obvodu a skupinu nepokrytých realizací testovaného obvodu,
- VI. vyhodnocení diagnostického příznaku (porovnáním výstupní hodnoty reálného testovaného obvodu s výstupní hodnotou realizace M_0 testovaného obvodu). Máme k dispozici dvě možná pokračování konstrukce (větve) lokalizačního testu, jednak pro hodnotu "+" a dále pro hodnotu "-". Pro přehlednost je vhodné výsledky zaznamenávat graficky, dostáváme tak tzv. **diagnostický strom**. Pro sestavení lokalizačního testu je třeba dokompletovat všechny větve diagnostického stromu,
- VII. výběr vhodného testovacího vektoru dalšího kroku lokalizačního testu. Postup výběru testovacího vektoru je ve dvou alternativách, podle hodnoty diagnostického příznaku, uveden níže:
 - i) pokud diagnostický příznak nabývá hodnoty "+", volíme další testovací vektor tak, aby částečně pokrýval momentálně nepokryté realizace testovaného obvodu. Vybíráme takovou vstupní kombinaci, která je schopna dílčí skupinu nepokrytých

realizací testovacího obvodu atomizovat na další, pokud možno stejně rozsáhlé, dílčí skupiny detekovatelných a nedetekovatelných realizací testovacího obvodu,

ii) pokud diagnostický příznak nabývá hodnoty "-", volíme další testovací vektor tak, aby částečně pokrýval momentálně pokryté realizace testovaného obvodu. Vybíráme takovou vstupní kombinaci, která je schopna dílčí skupinu pokrytých realizací testovacího obvodu atomizovat na další, pokud možno stejně rozsáhlé, dílčí skupiny detekovatelných a nedetekovatelných realizací testovacího obvodu,

iii) pokračujeme tak v analýze diagnostických příznaků, pokud lze najít takové vstupní kombinace, která jsou schopny atomizovat dílčí skupiny realizací testovacího obvodu,

VIII. bod VII. opakujeme pro všechny větve diagnostického stromu.

Detekční test získáváme minimalizací triviálního testu kombinačního obvodu. Lokalizační test pro kombinační obvody dostáváme vhodným řazením kroků detekčního testu. Pro jednoduché obvody s nižším počtem vstupů se minimalizace počtu kroků testu jeví jako neúčelná. Mez počtu vstupů je závislá na způsobu zápisu testů na vhodný nosič informací. Pokud použijeme běžný formát papíru je limitem řádově 15 vstupů. Pro větší počet vstupů je pak vhodnější provádět minimalizaci počtu kroků a tak zpřehlednit ověřování průceschopnosti testovaného obvodu. Minimalizace dosažená běžným postupem se blíží padesáti procentům při zachování stoprocentního diagnostického pokrytí.

Postup bude opět demonstrován v následujícím příkladu na jednoduchém kombinačním obvodu z příkladu č.1.9.

Příklad 1.10:

Sestavte lokalizační test obvodu z př. 1.9.

Postup:

test	A	B	C	Y	E_A	E_B	E_C	E_D	E_Y
T ₀	0	0	0	0			t_1	t_1	t_1
T ₁	0	0	1	1			t_0		t_0
T ₂	0	1	0	0	t_1		t_1	t_1	t_1
T ₃	0	1	1	1			t_0		t_0
T ₄	1	0	0	0		t_1	t_1	t_1	t_1
T ₅	1	0	1	1			t_0		t_0
T ₆	1	1	0	1	t_0	t_0		t_0	t_0
T ₇	1	1	1	1					t_0

A. pro další práci využijeme výsledek kroků I., II. a III. podle postupu vytvoření detekčního testu v příkladu č.1.9,

B. přiřazení realizací poruch obvodu detekovaným poruchám - krok IV. postupu sestavení lokalizačního testu,

označení	místo a typ poruchy	označení	místo a typ poruchy
M_1	At_0	M_6	Ct_1
M_2	At_1	M_7	Dt_0
M_3	Bt_0	M_8	Dt_1
M_4	Bt_1	M_9	Yt_0
M_5	Ct_0	M_{10}	Yt_1
M_0	bez poruchy		

test	A	B	C	Y	E_A	E_B	E_C	E_D	E_Y
T_0	0	0	0	0			M_6	M_8	M_{10}
T_1	0	0	1	1			M_5		M_9
T_2	0	1	0	0	M_2		M_6	M_8	M_{10}
T_3	0	1	1	1			M_5		M_9
T_4	1	0	0	0		M_4	M_6	M_8	M_{10}
T_5	1	0	1	1			M_5		M_9
T_6	1	1	0	1	M_1	M_3		M_7	M_9
T_7	1	1	1	1					M_9

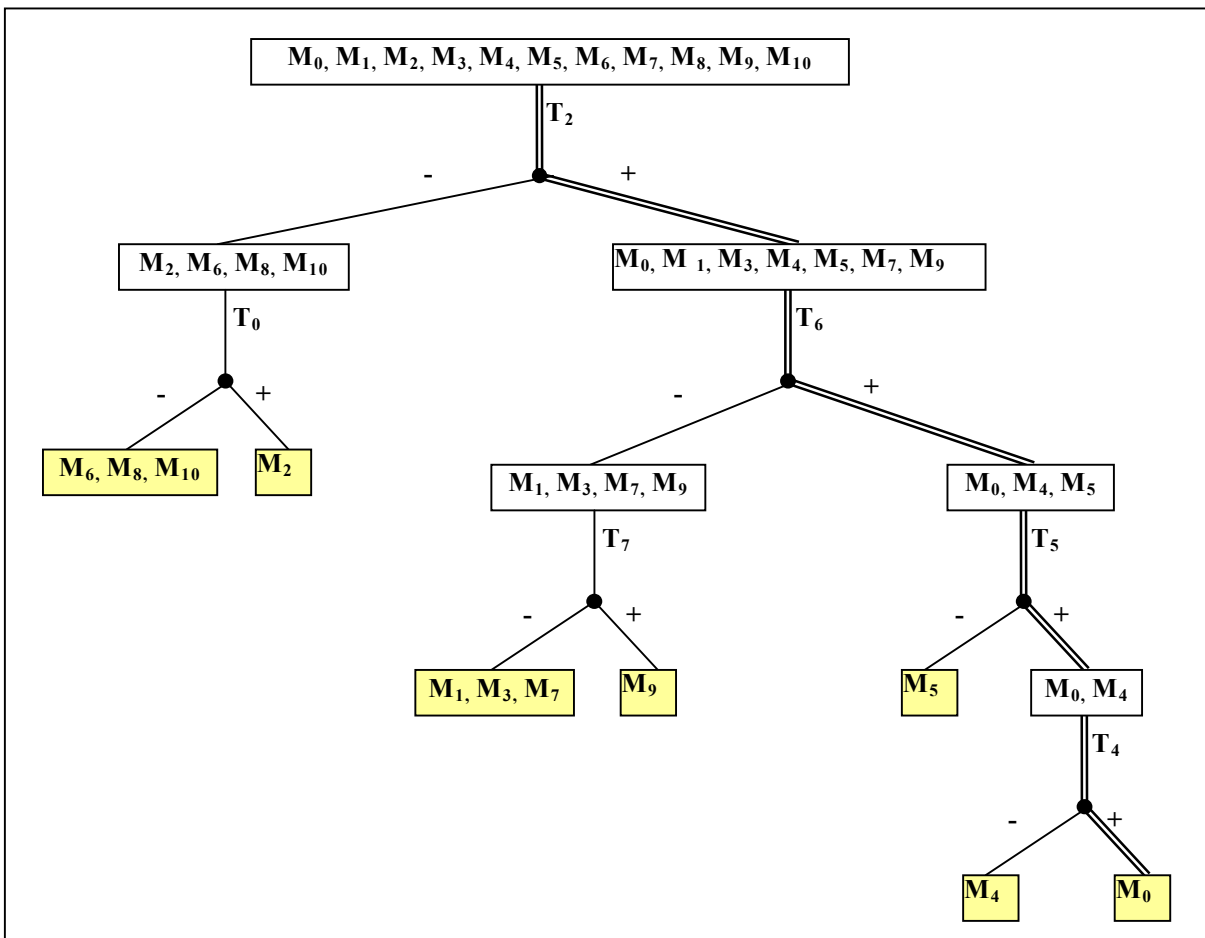
C. výběr testovacího vektoru metodou půlení intervalu - krok V. postupu sestavení lokalizačního testu. Jako první testovací vektor vybereme vstupní kombinaci T_2 . Množina všech možných realizací testovacího obvodu se podle diagnostického příznaku rozdělila na dvě skupiny. První skupina jsou detekovatelné realizace M_2, M_6, M_8, M_{10} . Druhou skupinou jsou nedetekovatelné realizace $M_0, M_1, M_3, M_4, M_5, M_7, M_9$. Pro každou skupinu musíme dále postupovat samostatně,

D. z tabulky získané v bodě b) tohoto příkladu se pro skupinu realizací M_2, M_6, M_8, M_{10} nabízí vybrat jako testovací vektor vstupní kombinaci T_0 . Pro ni jsou detekovatelné realizace M_6, M_8, M_{10} . Po vyhodnocení diagnostického příznaku získáváme jednu dílčí skupinu nedetekovatelné realizace M_2 , která představuje koncový vrchol lokalizačního testu. Druhou dílčí skupinu tvoří detekovatelné realizace M_6, M_8, M_{10} tohoto kroku,

E. další krok lokalizačního testu již nelze nalézt, protože žádná vstupní kombinace testovacího obvodu není schopna atomizovat dílčí skupinu M_6, M_8, M_{10} realizací testovacího obvodu,

F. obdobně postupujeme pro skupinu nedetekovatelných realizací $M_0, M_1, M_3, M_4, M_5, M_7, M_9$ z kroku c) řešení této úlohy. V této fázi sestavení kroku testu lze pro atomizaci z tabulky vybrat vstupní

kombinaci T_6 . Ta atomizuje skupinu na dílčí skupinu detekovatelných realizací M_1, M_3, M_7, M_9 a dílčí skupinu nedetekovatelných realizací M_0, M_4, M_5 testovacího obvodu,



Obr.č.1.13: Diagnostický strom lokalizačního testu

G. takto se postupuje až do vyčerpání všech možností atomizace dílčích skupin realizací. Výsledek je zachycen na schéma uvedeném na obr. č. 1.13. Zvýrazněně jsou vybarveny vrcholy lokalizačního testu. Z obrázku lze vypožorovat, že diagnostický strom poskytuje návod pro volbu jednotlivých testovacích vektorů v závislosti na hodnotě diagnostického příznaku. Vrchol diagnostického stromu představuje neprázdnou množinu realizací testovacího obvodu, které vykazují totožnou reakci na příslušnou posloupnost testovacích vektorů. Podle seznamu z bodu 4) postupu pro sestavení lokalizačního testu najdeme poruchy, které se v reálném prověřovaném obvodu mohou vyskytovat.

Řešení:

Na diagnostickém stromu lze vypožorovat, že posloupnost vstupních kombinací T_2, T_6, T_5 a T_4 - vyznačeno dvojitou čarou - tvoří detekční test obvodu z předchozí kapitoly (jedná se o realizaci M_0 testovacího obvodu).

1.5 Příznaková analýza

Struktura soudobých číslicových systémů je tvořena především sběrníkovými strukturami sestavených z LSI obvodů (např. mikroprocesorů, ROM, RAM, komplexních interfejsových jednotek, programovatelných struktur).

Zpracování a sběr dat, realizace algoritmických vztahů mezi nimi a komunikace mezi jednotlivými etapami jejich zpracování je prováděno především programovými prostředky. Technické vybavení slouží především k řízení komunikace a algoritmických vztahů mezi sběrníkovými jednotkami. Pro tyto systémy je charakteristický tok dat systémem. Jeho funkční a operační charakteristiky nejsou nezbytně spjaty s danou hardwarovou strukturou a je obtížné je charakterizovat a definovat. Další charakteristická vlastnost soudobých elektronických systémů spočívá v tom, že porty (brány) obvodů jsou obousměrné (tj. mají na téže svorce vstup i výstup).

Na rozdíl od dříve běžně používaných kombinačních a sekvenčních obvodů je hledání a lokalizace poruch uvnitř mikroprocesoru - a tudíž i těch nejjednodušších mikroprocesorových systémů - obtížné i přes podrobnou dokumentaci. Proto je většina oprav mikroprocesorových částí přístrojů řešena výměnou celé desky.

V jiné situaci jsou vývojáři elektronických systémů. I když jsou vyzbrojeni logickými analyzátory a další speciální přístrojovou technikou, vyžaduje nalezení příčiny neuspokojivé funkce mikroprocesorového systému se sběrníkovou strukturou detailní znalost obvodů, řešeného algoritmu a programovacího nástroje pro sestavení algoritmu. Nalezení občas se vyskytujícího nežádoucího chování systému může zabrat velmi mnoho času. Hlavní problém je v tom, že testovací přístroje poskytují buď příliš mnoho nebo příliš málo informací. Soudobé přístroje poskytují mnoho možností konfigurací pro sledování podmínek chování systému a často se stává, že diagnostik musí hledat vhodnou konfiguraci podmínek pro nalezení chyby v návrhu elektronického systému.

Pro efektivní nalezení chybně pracující části elektronického systému se kromě jiných technik naskytá možnost komprese vnitřních dat. Dlouhé datové řetězce, které se cyklicky opakují v existujícím systému běžícími normální operační rychlostí, je možno komprimovat do kompaktního, snadno interpretovatelného a charakteristického výrazu. Tento výraz nemusí říkat nic jiného, než že příslušný zkoumaný uzel pracuje nebo nepracuje správně. Tohoto principu používá příznaková analýza elektronických systémů.

1.5.1 Základ příznakové analýzy

Příznaková analýza představuje výkonný nástroj pro prověřování průceschopnosti elektronických systémů.

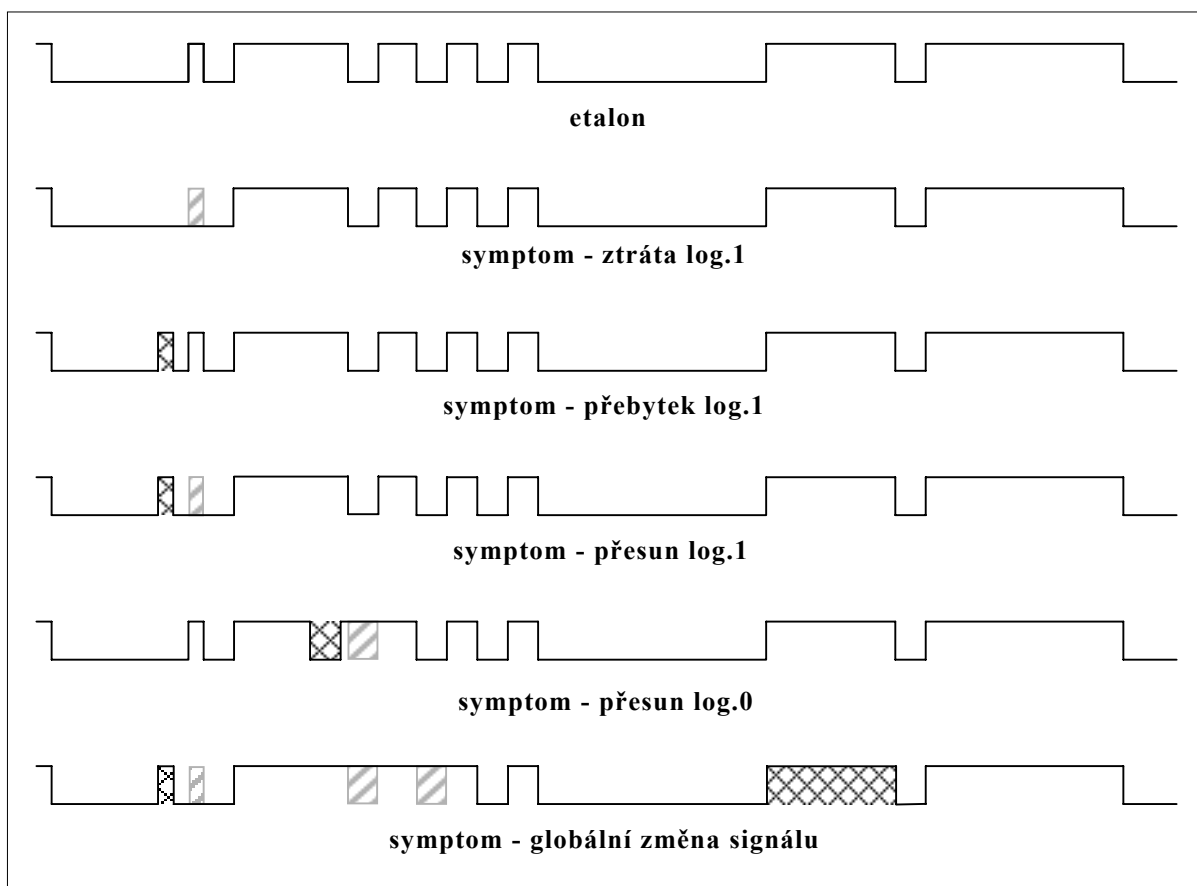
Princip činnosti příznakové analýzy je následující. Každý uzel elektronického systému je chápán jako zdroj sekvenčního signálu. Za určitý čas lze v tomto bodě vysledovat výskyt určitého počtu logických nul (log. 0) a logických jedniček (log. 1). Pokud se podaří zajistit (identifikovat) opakovatelnou synchronizaci tohoto signálu, je možné jej snadno, opakovatelně a věrohodně zapsat do sériového registru a poté nad ním provádět další operace. Signál zapsaný do registru můžeme chápat jako **kód** a je třeba najít vhodný nástroj pro ověření jeho **autenticity**. Tu lze ověřit porovnáním aktuálního kódu s kódem **etalonovým**, který získáme měřením na bezporuchovém elektronickém systému.

Základním předpokladem úspěšné aplikace příznakové analýzy je skutečnost, že kód generovaný bezchybným obvodem a představovaný posloupností **log.0** a **log.1**, se liší od kódu ovlivňovaného poruchou obvodu - od poruchou modifikované posloupnosti **log.0** a **log.1**. Základem příznakové analýzy je tedy, jako u všech diagnostických metod, umění přinutit poruchu ovlivnit chování signálu v některém uzlu systému, který je přístupný pro sondy měřicích přístrojů. Elegance příznakové analýzy pak spočívá v prostém porovnávání **příznaků** kódu v uzlech správně fungujícího elektronického systému (**etalon**) s příznaky kódu v uzlech elektronického systému s poruchou (**symptom**). Při použití příznakové analýzy není třeba analyzovat chování obvodu a pronikat do podstaty vzniku nežádoucích posloupností signálů. Z tohoto je zřejmé, že příznaková analýza je vhodnou metodou především pro opravy již navržených a dříve správně fungujících elektronických systémů.

Autentizace kódu uchovaného v záznamovém registru může být pro rozsáhlé posloupnosti poněkud náročná na kapacity diagnostického systému. Pro zjednodušení autentizace se proto v příznakové analýze používá **komprese dat**. Principů komprese je obecně více. Nejjednodušší je zjišťovat počet **log.1** v symptomu - viz obr. č. 1.14.

Z obrázku 1.14 lze vypožorovat, že pouhé sečítání počtu **log.1** nemusí být vždy úspěšné, protože speciálně u symptomů přesun **log.1** nebo přesun **log.0** vede součet k chybné autentizaci. K eliminaci tohoto jevu se používá mechanismu komprese dat. Jako zcela vyhovující lze použít kontrolní součet. Kompresní mechanismus signál v testovaném uzlu interpretuje jako sériově vysílaná data a tato zamění posloupností bitů, tzv. zabezpečovacím slovem - **příznakem**. Příznak je generován vytvářecím polynomem. Jedná se o mechanismus běžně používaný pro zabezpečení sériově přenášené číselné informace. V běžných realizacích

se předpokládá délka kódu 2^{16} bitů. Tato délka slova vyplývá z doposud nepoužívanější šířky sběrnice běžných mikroprocesorových systémů, které používají šestnáctibitové sběrnice. Struktura vytvářecího polynomu je dána požadovanými vlastnostmi příznaku. Základním požadavkem je, aby dva nebo více různých signálů nemělo stejný příznak.



Obr. č. 1.14: Příklady projevů poruch v symptomech testovaného obvodu

Jednoznačnou identifikaci vstupního kódu příznakové analýzy zajišťuje synchronizační signál - **CLK** a **časové okno** definované hranami signálů **START** a **STOP**. Synchronizační signál definuje platné hodnoty testovaného signálu (vstupního signálu příznakového analyzátoru) označovaného termínem **DATA**. Časové okno definuje část cyklicky opakované činnosti testovaného systému podrobené diagnostické analýze a v příznakové analýze je někdy označováno termínem **interval START - STOP**. Většina číslicových systémů je konstruována jako systémy synchronní a tudíž změny signálů v uzlech testovaného obvodu jsou synchronizovány některou z hran synchronizačního signálu.

Pro přesnou definici časového okna tedy nestačí pouze vybrat signály definující jeho začátek a konec, ale je nutno definovat hrany těchto signálů. Totéž platí pro volbu okamžiků vzorkování signálu **DATA**. Musí se zvolit hrana synchronizačního signálu, která nezpůsobuje

změnu měřených signálů. Je žádoucí zvolit inverzní hranu synchronizačního signálu, která vyjadřuje okamžik odeznění přechodových stavů a ustálení amplitud měřených signálů. Takto je možno jednoznačně definovat podmínky testování číslicového systému podrobeného příznakové analýze a podmínky pro jednoznačnou a opakovatelnou posloupnost diagnostických operací umožňujících lokalizaci poruchy testovaného číslicového systému.

Za předpokladu splnění podmínek definovaných v předchozím odstavci vstupují **DATA** do příznakového analyzátoru synchronně s hodinovým signálem testovaného přístroje po dobu **intervalu START-STOP**. Jak synchronizace, tak časové okno je definováno testovaným systémem. Pro generování příznaku slouží posuvný registr doplněný obvody realizujícími funkci vytvářejícího polynomu. Malé kódové vzdálenosti údajů signálu **DATA** kódu jsou příznakem interpretovány jako velké kódové vzdálenosti, jinak řečeno i drobné změny vstupního signálu se projeví zásadní změnou příznaku.

1.5.2 Generování příznaku

V příznakových analyzátorech se, po určitém vývoji, ustálilo používat ve funkci příznaku zbytek po dělení symptomu o délce **m-bitů** (je tedy představován polynomem **m-1 stupně**) vytvářejícím polynomem **n-tého stupně**. Nejschůdnější realizaci dělení představuje lineární zpětnovazební posuvný registr. Zbytek po dělení, uložený v registru po **m** krocích sériově zaváděného symptomu, je tedy produktem komprimace - hledaným příznakem.

Obecně platí, že při dělení vzniká ztráta informace. Ztráta se může projevit tak, že mohou existovat třídy kódů (ekvivalence) nerozlišitelné podle svých příznaků. Lze odvodit tvary signálů **DATA**, které dávají stejné zbytky po dělení vytvářejícím polynomem **P(x)** **n-tého** stupně. Pokud má signál **DATA** délku **m** bitů, pak je reprezentován polynomem stupně **(m-1)** a může mít kromě své aktuální prezentace celkem **(2^{m-1}-1)** dalších interpretací, které vlastně představují symptomy. Podíl signálu **DATA** a vytvářejícího polynomu pak může nabýt tvaru polynomu stupně **(m-1)-n** nebo menšího, přičemž ke každému podílu přísluší zbytek. Jestliže stupeň vytvářejícího polynomu je menší než stupeň polynomu představujícího signál **DATA**, existuje **(2^{(m-1)-n}-1)** nenulových násobků a tedy **(2^{(m-1)-n}-1)** reprezentací signálu **DATA** se stejným zbytkem. Každý nejednoznačně detekovatelný signál **DATA** je jiným násobkem vytvářejícího polynomu. Následující tabulka zobrazuje tvary polynomů, které mají stejný zbytek po dělení vytvářejícím polynomem **P(x)**. Obvodové řešení je možné pomocí několika variant zpětnovazebních registrů

Zbytek signály DATA (vstupní polynomy příznakového analyzátoru)				
0	P(x)	x•P(x)	(x ^{(m-1)-n} +...+x+1)•P(x)
1	P(x)+1	x•P(x)+1	(x ^{(m-1)-n} +...+x+1)•P(x)+1
.	
.	
x ⁿ⁻¹ + + x + 1	P(x) + (x ⁿ⁻¹ + + x + 1)	x•P(x) + (x ⁿ⁻¹ + + x + 1)	(x ^{(m-1)-n} + ... + x + 1)•P(x) + (x ⁿ⁻¹ + + x + 1)

Tab.1.4: Tvary vstupních polynomů se stejným zbytkem po dělení

Ve smyslu předchozí tabulky lze pravděpodobnost výskytu nejednoznačně detekovatelných signálů **DATA** vyjádřit následujícím vztahem:

$$P_1 = \frac{2^{(m-1)-n}-1}{2^{(m-1)}-1} \approx 2^{-n} \quad \text{platí pro } m-n \gg 1$$

Ve výše uvedeném vztahu jsou obsaženy i primitivní tvary vytvářejícího polynomu a to x^n (registr bez zpětné vazby) a x^n+1 (registr pouze s jednou zpětnou vazbou přes celou svou délku) pro něž platí, že příznakem je právě posledních n bitů signálu **DATA**.

V případě že $m < n$ budou rozlišeny všechny syndromy za předpokladu, že se porucha projeví v m bitech signálu **DATA**. Toto lze v praktických úlohách splnit - např. při testování paměti ROM s řídicím programem je adresována pouze ta část adresového prostoru, do něhož je paměť ROM mapována.

Doposud byla popisována obecná situace výskytu chyby v kódu snímaném v uzlu měřeného obvodu. V takových případech se změni hodnota jednoho nebo více bitů, přičemž mezi nimi není žádná souvislost (kauzalita). V reálných obvodech číslicových zařízení se konkrétní porucha v cyklu činnosti systému projeví opakovaně. Jedná se o takové činnosti jako přenos po sběrnici, generování doplňku při sčítání, paralelní násobení a dělení, kódování a šifrování dat a pod.

Příklad realizace lineárního zpětnovazebního posuvného registru pro dělení polynomu $x^7+x^6+x^5+x^4+x^2+1$ polynomem $x^5+x^4+x^2+1$.

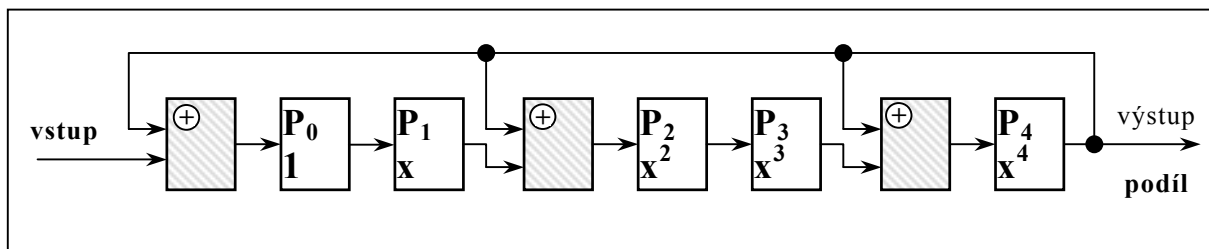
Aritmetické vyjádření dělení:

$$\begin{array}{r}
 (x^7 + x^6 + x^5 + x^4 + x^2 + 1) : (x^5 + x^4 + x^2 + 1) = x^2 + 1 \\
 \underline{-x^7 - x^6} \qquad \qquad \underline{-x^4 - x^2} \\
 \qquad \qquad \qquad + x^5 \qquad \qquad \qquad + 1 \\
 \qquad \qquad \qquad \underline{-x^5 - x^4 - x^2 - 1} \\
 \qquad \qquad \qquad \qquad \qquad \underline{-x^4 - x^2}
 \end{array}$$

Podíl = $+x^2 + 1$, tedy binárně **101**

Zbytek = $-(x^4 + x^2)$, jeho binární interpretace - **10100**

Jedna z více možností realizace dělení lineárním zpětnovazebním posuvným registrem je uvedena na následujícím schéma.



Postup dělení je znázorněn v následující tabulce.

vstupní posloupnost								obsah registru					výstup		
V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	P ₀	P ₁	P ₂	P ₃	P ₄	O ₀	O ₁	O ₂
1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0
	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0
		1	0	1	0	1	1	1	1	0	0	0	0	0	0
			1	0	1	0	1	1	1	1	0	0	0	0	0
				1	0	1	0	0	1	1	1	1	0	0	0
					1	0	1	0	0	0	1	0	1	0	0
						1	1	0	0	0	0	1	0	1	0
							1	0	0	1	0	1	1	0	1

Maximální délku cyklu může mít více polynomů stejného stupně. Z hlediska optimalizace konstrukce příznakového analyzátoru je vhodné zvolit polynom s minimálním počtem zpětných vazeb. Nejčastěji používaný tvar polynomu má tvar:

$$P(x) = x^{16} + x^{12} + x^9 + x^7 + 1$$

Do příznakového analyzátoru vstupuje signál DATA buď v přímém nebo invertovaném tvaru a představuje slovo o délce šestnácti bitů poskytujících celkem 2^{16} stavů. Posuvný registr tedy může během měřicího intervalu projít celkem až 65 536 stavy (což je definováno délkou časového okna). Tyto stavy jsou pak kódovány, vzniká příznak interpretovaný jako čtyři hexadecimální číslice. Pro přímé pozorování bývá vysvícen na sedmsegmentových indikátorech nebo jiných vhodných výstupních zařízeních. Příznak je charakteristické číslo, reprezentující časově závislou logickou aktivitu příslušného uzlu (měřicího bodu) během daného měřicího intervalu. Jakákoliv změna v chování signálu v tomto uzlu (např. posune-li se hrana o jeden hodinový cyklus dopředu nebo dozadu) se projeví změnou příznaku, což indikuje chybu ve funkci testovaného obvodu. Výhodou použitého algoritmu komprese je to, že i měřicí intervaly přesahující 2^{16} hodinových cyklů dávají stále platné opakovatelné příznaky.

Signál, který budí testovaný uzel se nazývá „stimul“. U příznakové analýzy je zdrojem stimulu sám testovaný systém. Tato skutečnost

podporuje možnost testovat elektronická zařízení při jejich plné funkční rychlosti a to i tak, že vybrané obvodové celky budou testovány nezávisle na zbytku systému. Rovněž synchronizaci a měřicí interval pro příznakovou analýzu je žádoucí odvodit od činnosti testovaného systému. V mikroprocesorovém systému může být stimulem vhodně strukturovaný program (obvykle v paměti mikroprocesorového systému), který uvede do činnosti testované části systému. Vezmeme-li v úvahu obrovské možnosti mikroprocesoru při manipulaci s daty, není vytvoření vhodných posloupností stimulů pro vybuzení všech bloků číslicového systému příliš obtížné.

Příznaková analýza je efektivní pro složité a rozsáhlé číslicové systémy. Pro efektivní činnost příznakové analýzy je třeba vnitřní uzly systému vybudit tak, aby průběžně měnily svůj stav. Nemění-li uzel bezchybného systému svůj stav, poskytuje velmi omezenou diagnostickou informaci – není zcela vyzkoušen. Pro elektronické systémy je charakteristické, že konstantní logickou úroveň v uzlu způsobuje stálá porucha.

Ve skutečnosti příliš nezáleží na tom jaké funkce uzlu elektronického systému vybudí k poskytování diagnostické informace. Úplná pravdivostní tabulka je sice zdrojem exaktních údajů o předpokládaném chování testovaného obvodu a může detekovat více chyb, ovšem za cenu použití podstatně rozsáhlejších prostředků a časových nákladů na její vytvoření a prověření na testovaném zařízení. Neobejdeme se bez specializovaných přístrojů a konstrukčních úprav elektronického systému nebo diagnostických vestaveb, které rozšiřují technické a programové vybavení systému a obecně tak snižují jeho pravděpodobnost bezporuchového provozu.

V běžných mikroprocesorových systémech se pro detekci bezporuchového stavu testovaného systému používá postup **nabalování**. Napřed se otestuje nezbytné technické vybavení testovaného systému. Tato část technického vybavení se nazývá **tvrdé jádro systému**. Pokud je tvrdé jádro funkčně vyhovující, je metodou nabalování postupně otestováno zbývající technické vybavení testovaného systému.

Tvrdé jádro obvykle představuje základní technické vybavení umožňující testovat další funkční bloky testovaného systému. Nejčastěji představuje bezchybné napájení, spolehlivý a stabilní generátor synchronizace systému, obvody výběru následující aktivity a obvody adresování paměti s řídicím programem - **MONITOREM**.

Nejjednodušší realizace výše uvedených požadavků představuje tzv. „volný běh“. Volný běh znamená uvedení testovaného systému do činnosti v opakující se smyčce s minimálním počtem aktivizovaných

prvků testovaného systému na jedné straně, ale s maximálním počtem aktivně vybuzených uzlů na straně druhé.

U mikroprocesorových systémů lze volný běh vytvořit rozpojením datové (případně sdílené) sběrnice a mikroprocesoru vnutíme takovou instrukci, která způsobí plynulé cyklické adresování celého adresového prostoru operační paměti. Nejčastěji se pro tuto činnost používá instrukce **NOP (NO OPERATION)** nebo vhodná registrová instrukce, např. zvětšování obsahu registru akumulátoru o **1**. Kód instrukce NOP bývá u celé řady mikroprocesorů představován hexadecimálním kódem **00**. Registrové operace jsou naopak interpretovány kódem **FF** nebo nějakým blízkým kódem. Obě instrukce mají většinou zanedbatelnou délku trvání fáze EXECUTE a navenek mikroprocesoru se projevují pouze fází FETCH (změna obsahu registru PC, který obsahuje adresu následující instrukce v operační paměti).

Sledováním příznaků testovaného systému ve volném běhu lze ověřit správnou funkci jeho jádra, většinu kombinační logiky adresových a řídicích linek (hlavně dekodéry a obsah paměti ROM) a funkci datové sběrnice. Volný běh sám o sobě nemůže dostatečně vyzkoušet všechny obvody systému jako RAM a většinu obvodů typu DMA a přerušení, případně podřízené mikroprocesory. Pro vyzkoušení této třídy obvodových prvků musí být generovány testovací stimuly, které obsahují únosné množství budících podnětů. Často bývají tyto stimuly obsaženy v programu samokontroly číslicového systému jako podprogramy.

Literatura

- [1] Z.Burian, V. Třeštíková. *Diagnostika v mikroelektronice*. Praha, ČVUT 1988.
- [2] J.Hlavička. *Diagnostika a spolehlivost číslicových systémů*. Praha, ČVUT 1978.
- [3] J. Sýkora. *Diagnostika telekomunikačních zařízení*. Praha, ČVUT 1991.
- [4] J. Hlavička, E. Kottek, J. Zelený. *Diagnostika elektronických obvodů*. Praha, SNTL/ALFA, 1981.
- [5] H. K. Reghbatl. *Testing & Validation Techniques*. Washington, EEE Computer Society Press / North-Holland, 1985
- [6] J. Bednařík. *Technika spolehlivosti v elektronické praxi*. Praha, SNTL 1990
- [7] R. Leitl. *Spolehlivost elektronických systémů*. Praha, SNTL 1990
- [8] M. Novák, V. Přenosil, M. Svítek, Z. Votruba. *Problémy spolehlivosti, životnosti a bezpečnosti systémů*. Praha, Neural Network World 2005
- [9] Military Handbook MIL-HDBK-217F: *Reliability prediction of electronic equipment*.
- [10] PETRŽELKA, B. - PŘENOSIL, V.: *Konstrukce a architektura číslicových počítačů*. (Vysokoškolská učebnice U-8062/2, UO). Brno, UO 2005