

# Spolehlivost a diagnostika číslicových systémů

## Testování sekvenčních obvodů, Boundary-Scan

studijní text frekventantů předmětu PV 171/11

### Obsah

11 Testování sekvenčních obvodů .....	2
11.1 Nastavitelnost a pozorovatelnost obecného sekvenčního obvodu .....	3
11.1.1 Test sekvenčního obvodu .....	6
11.2 Technická řešení nastavitelnosti a pozorovatelnosti .....	7
11.2.1 Rozpojení zpětné vazby sekvenčního obvodu .....	7
11.2.2 Sériový diagnostický přístup k paměti vnitřního stavu .....	8
11.2.3 Přímý diagnostický přístup k paměti vnitřního stavu .....	11
11.2.4 Standard Boundary Scan .....	13
11.2.4.1 Princip činnosti metody Boundary-Scan .....	14
11.2.4.2 Řadič Test Access Port (TAP) .....	17
11.2.4.3 JTAG konektory .....	17
11.2.4.4 Instrukce systému Boundary-Scan .....	18
11.2.4.5 Testování integrovaných obvodů .....	18
11.2.4.6 Interní testování integrovaných obvodů .....	19
11.2.4.7 Testování na úrovni desek .....	21
Seznam obrázků .....	22
Literatura .....	23

## 11 Testování sekvenčních obvodů

Pro sekvenční logické obvody platí, že výstupní vektor závisí nejen na vstupním vektoru, ale i na *sekvenci předchozích vstupních vektorů*. V závislosti na předcházejících vstupních vektorech může sekvenční systém při určitém vstupním vektoru generovat různé výstupní vektory. To znamená, že systém si musí pamatovat předešlé situace a musí mít tedy *paměť* (aktuální výstup závisí na historii systému). Informace o stavu této paměti dávají vnitřní proměnné systému. Chování sekvenčního logického systému lze popsat funkcí:

$$F_s : X^* \rightarrow Z^*$$

kde  $X^*$ ,  $Z^*$  jsou nekonečné množiny všech posloupností s konečnou délkou sestavené ze vstupních vektorů  $x_i$  množiny  $X$  resp. výstupních vektorů  $z_i$  množiny  $Z$ .  $X^*$  a  $Z^*$  nazýváme vstupní a výstupní slovo. Funkce  $F_s$  každému vstupnímu slovu  $X^*$  přiřazuje výstupní slovo  $Z^*$ , které má vždy stejnou délku. Vstupní slovo  $X^*$  je posloupnost vektorů  $x$  od časového okamžiku  $t = 0$ , při kterém se systém nacházel ve stavu  $s$ . Funkce  $F_s$  se nazývá funkcí sekvenčního systému (sekvenční zobrazení). Funkce sekvenčního obvodu je popsána dvojicí logických rovnic:

$$\begin{aligned} s_{i+1} &= f_s(x_i, s_i) \\ z_i &= f_z(x_i, s_i) \end{aligned}$$

Význam symbolů:

$f_s$  - funkce následného stavu systému - přechodová funkce,

$f_z$  - výstupní funkce systému,

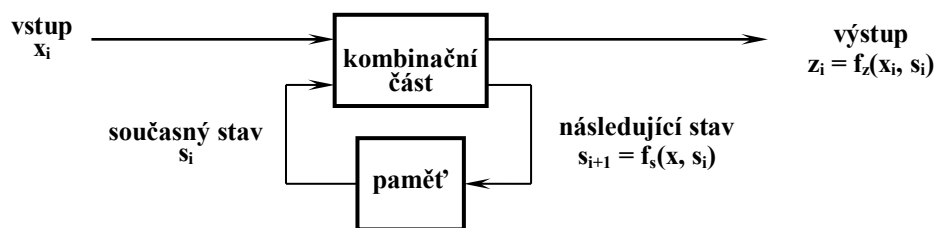
$x_i$  - aktuální vstupní vektor systému z množiny  $X$ ,

$z_i$  - aktuální výstupní vektor systému z množiny  $Z$ ,

$s_i$  - současný stav systému z množiny  $S$ , který je dán okamžitým stavem vnitřních proměnných,

$S$  - množina všech možných vnitřních stavů systému - odpovídá míře informace pamatované systémem,

$i$  - pořadová hodnota kroku diskrétního času



Obrázek 11.1: Blokové schéma sekvenčního obvodu

Pro ověření správné činnosti sekvenčního obvodu potřebujeme prověřit jak správnost činnosti funkce generování výstupních proměnných (tedy funkci  $f_z$ ), tak správnost generování proměnných reprezentujících následující stav sekvenčního obvodu (tedy funkce  $f_s$ ).

Metody testování obou funkcí sekvenčního obvodu se shodují s testováním funkce kombinačních obvodů. Základním problémem je však nepřístupnost výstupů funkce  $f_s$  a nepřístupnost výstupů paměti vnitřního stavu sekvenčního obvodu, přičemž dílčí úlohou testování je zjistit správnost generování posloupností vnitřního stavu v závislosti na posloupnostech vstupních vektorů.

V případě obecného sekvenčního obvodu můžeme na jeho vnitřní stav usuzovat pouze podle

chování výstupu, přičemž v každém kroku testu aktuální vnitřní stav ovlivňuje výstupní proměnné a přechod do následujícího vnitřního stavu. Pro potřeby diskuse nad vlastnostmi obecného sekvenčního obvodu jsou zavedeny termíny *nastavitelnost* a *pozorovatelnost*. Nastavitelnost představuje vlastnost sekvenčního obvodu, která umožňuje nastavit požadovaný vnitřní stav sekvenčního obvodu. Pozorovatelnost pak představuje vlastnost umožňující zjistit aktuální vnitřní stav sekvenčního obvodu.

Řešení úlohy nastavitelnosti a pozorovatelnosti u obecného sekvenčního obvodu není triviální úlohou a v obecném případě nemusí mít řešení. Napřed si probereme problematiku nastavitelnosti a pozorovatelnosti na jednoduchých příkladech funkce obecného sekvenčního obvodu a poté bude diskutována problematika konstrukčních úprav poskytujících možnost implementace nastavitelnosti a pozorovatelnosti sekvenčního obvodu.

### 11.1 Nastavitelnost a pozorovatelnost obecného sekvenčního obvodu

Základním problémem souvisejícím s testováním správné činnosti sekvenčních obvod je vnitřní stav po zapnutí napájení. Nastavení jednotlivých klopných obvodů paměti vnitřního stavu je náhodnou záležitostí. Výchozí vnitřní stav jednotlivých fyzických realizací téhož obvodu je odlišný, a proto je relevantní předpokládat, že počáteční vnitřní stav obecného sekvenčního obvodu je nedefinovaný a před vlastním testováním je třeba vnitřní stav buď zjistit, nebo vhodným postupem nastavit vhodnou posloupností vektorů vstupních proměnných. Nabízejí se dvě základní možnosti, použít:

- *synchronizační posloupnost* - posloupnost vektorů vstupní proměnné, která uvede sekvenční obvod do předem definovaného stavu z jakéhokoliv jiného vnitřního stavu,
- *rozlišovací posloupnosti* - posloupnost vektorů vstupní proměnné, která umožňuje zjistit, v jakém vnitřním stavu se sekvenční obvod nacházel před aplikací této posloupnosti a v jakém vnitřním stavu se nachází po aplikaci této posloupnosti.

Po zjištění aktuálního vnitřního stavu sekvenčního obvodu je možné zahájit testování jeho funkčnosti. Vzhledem k tomu, že u běžného sekvenčního obvodu nemáme možnost přístupu do vnitřní struktury, nabízí se pro jeho otestování následující postup - na základě znalosti aktuálního vnitřního stavu přivedeme na vstupy sekvenčního obvodu posloupnost vektorů vstupních proměnných a sledováním výstupu ověřit, že sekvenční obvod plní požadovaný algoritmus. Na správnost fungování funkce  $f_s$  a paměti vnitřního stavu usuzujeme prostřednictvím výstupní funkce  $f_z$ . Problémem je, že porucha může být i v obvodech generování funkce  $f_z$  a tudíž tento postup umožňuje realizovat pouze detekční (go - no-go) testy.

Druhou možností je ověřovat posloupnosti nastavování vnitřních stavů obecného sekvenčního obvodu. K tomu je třeba mít k dispozici *nastavovací posloupnost*, pomocí níž obvod uvedeme do požadovaného vnitřního stavu a poté *rozlišovací posloupnosti* ověřit, zda dosažený vnitřní stav sekvenčního obvodu souhlasí s vnitřním stavem nastavovaným správně fungující funkcí  $f_s$  a správně fungující pamětí vnitřního stavu. Na správnost fungování funkce  $f_s$  a paměti vnitřního stavu usuzujeme opět prostřednictvím výstupní funkce  $f_z$ . Dvojice *nastavovací posloupnost* a *rozlišovací posloupnost* tvoří elementární krok *kontrolní posloupnosti* obecného sekvenčního obvodu. Pro testování obecného sekvenčního obvodu musíme mít tedy k dispozici:

- *nastavovací posloupnost* - posloupnost vektorů vstupní proměnné, která nastavuje požadovaný vnitřní stav sekvenčního obvodu,
- *kontrolní posloupnosti* - posloupnost vektorů vstupní proměnné prověřující správnou

činnost sekvenčního obvodu. Je zřejmé, že součástí kontrolní posloupnosti jsou nastavovací a rozlišovací posloupnosti.

Při testování můžeme postupovat krok po kroku - pak hovoříme o elementární *granularitě* testování. Elementární krok kontrolní posloupnosti můžeme rozšířit o množinu vstupních vektorů vstupní proměnné a tím zvětšit krok testování. Postup je následující:

- nastavení sekvenčního obvodu do ***n-tého*** (výchozího) vnitřního stavu,
- převedení posloupnosti ***k*** vektorů vstupních proměnných na vstupy sekvenčního obvodu s cílem uvedení sekvenčního obvodu do ***(n+k)-tého*** (průběžného) stavu,
- rozlišovací posloupností identifikovat aktuální vnitřní stav sekvenčního obvodu.

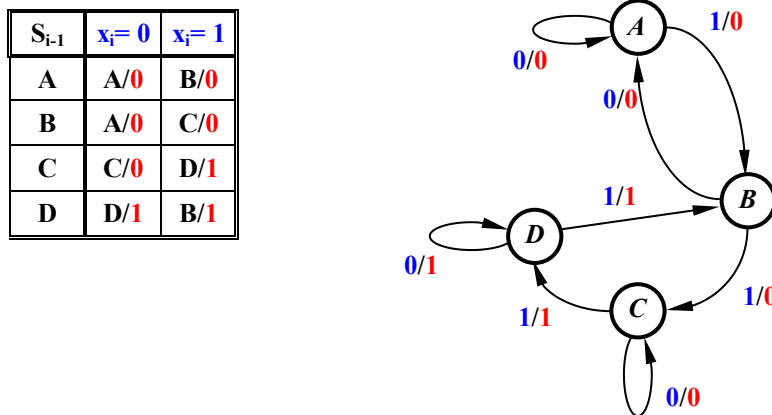
V tomto případě je granularita testování rovna hodnotě ***k***. a hovoříme o ***k-té třídě*** kroku kontrolní posloupnosti sekvenčního obvodu.

Základním problémem testování sekvenčních obvodů je skutečnost, že ne vždy je možné najít požadované posloupnosti vektorů vstupních proměnných a že na vnitřní stav testovaného sekvenčního obvodu usuzujeme nepřímo na základě chování výstupu.

Problematika nalezení rozlišovací a synchronizační posloupnosti je uvedena v následujících příkladech.

### **Příklad 11.1**

*Sestrojte rozlišovací posloupnost pro sekvenční obvod popsany níže uvedenou tabulkou a grafem přechodů. Výstupní proměnná sekvenčního obvodu je proměnná  $z_i$ .*



Obrázek 11.2: Tabulka a graf přechodů sekvenčního obvodu.

### **Řešení:**

Úkolem je najít takové posloupnosti vstupního signálu  $x_i$ , které umožní sledováním výstupu jednoznačně detekovat počáteční vnitřní stav testovaného sekvenčního obvodu.

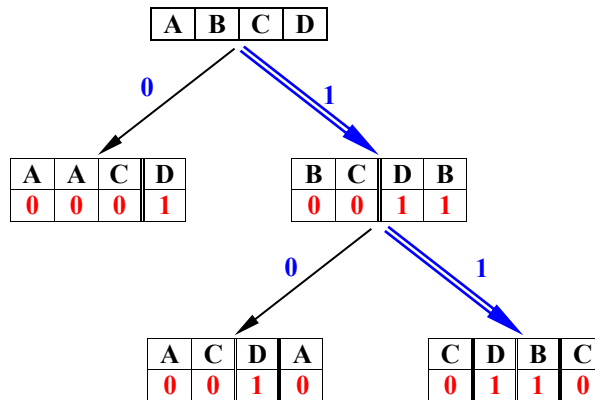
Protože na hodnotu vnitřní proměnné můžeme usuzovat pouze podle hodnot výstupní proměnné, hledáme pro každý aktuální vnitřní stav takovou hodnotu vstupní proměnné, která rozdělí množinu stejnolehlých následujících vnitřních stavů na dvě skupiny navzájem se odlišujícími hodnotou výstupní proměnné - na níže uvedeném schéma jsou skupiny odděleny středníkem.

Postup opakuje i pro takto získané skupiny - nyní aktuálních - vnitřních stavů, uvnitř nichž se opět snažíme najít hodnoty vstupní proměnné, které rozdělí stejnolehlé vnitřní stavy

skupin (v této fázi již následující vnitřní stavy) na další dvě podskupiny navzájem se odlišující hodnotou výstupní proměnné.

Dělení na podskupiny končí, jakmile jsou podskupiny tvořeny pouze jedním následujícím vnitřním stavem.

Postup je zachycen na následujícím obrázku.



Obrázek 11.3: Rozlišovací posloupnost sekvenčního obvodu

#### Výsledek:

Rozlišovací posloupnost sekvenčního obvodu je  $x_{1,2} = 1, 1$ . Počáteční vnitřní stav sekvenčního obvodu identifikuje výstupní posloupnost obvodu následujícím způsobem:

počáteční vnitřní stav **A** (koncový stav C), výstupní posloupnost je **0,0**

počáteční vnitřní stav **B** (koncový stav D), výstupní posloupnost je **0,1**

počáteční vnitřní stav **C** (koncový stav B), výstupní posloupnost je **1,1**

počáteční vnitřní stav **D** (koncový stav C), výstupní posloupnost je **1,0**

Z předchozího příkladu je zřejmé, že po provedení dvou kroků je známo v jakém výchozím vnitřním stavu se testovaný sekvenční obvod nacházel před aplikací rozlišovací posloupnosti a v jakém koncovém vnitřním stavu se nachází po její aplikaci.

Pro prověření činnosti tohoto obvodu máme nyní možnost přivádět na vstup takové posloupnosti vstupní proměnné, které prověří, zda testovaný obvod realizuje všechny přechody mezi vnitřními stavy. Na správnost funkce opět usuzujeme podle hodnot výstupní proměnné  $z_i$ .

Pokud byl testovaný sekvenční obvod ve výchozím vnitřním stavu **A**, pak po aplikaci rozlišovací posloupnosti je ve vnitřním stavu **C**. Nyní je možno obvod prověřit následující posloupností:

$$x_i = 1, 0, 1, 0, 0, 1, 1, 0$$

odpovědi správně fungujícího obvodu je následující výstupní posloupnost:

$$z_i = 1, 1, 1, 0, 0, 0, 0, 0$$

Problémem výše uvedeného postupu je skutečnost, že obvod po aplikaci rozlišovací posloupnosti může být ve více vnitřních stavech - v tomto případě ve stavech **C**, **D** nebo **B**. Pro každý z těchto vnitřních stavů je nutno hledat samostatnou testovací posloupnost. Z tohoto hlediska je výhodnější použít synchronizační posloupnost, která testovaný sekvenční obvod převede vždy do jednoho vnitřního stavu a pro otestování obvodu stačí pouze jedna

testovací posloupnost.

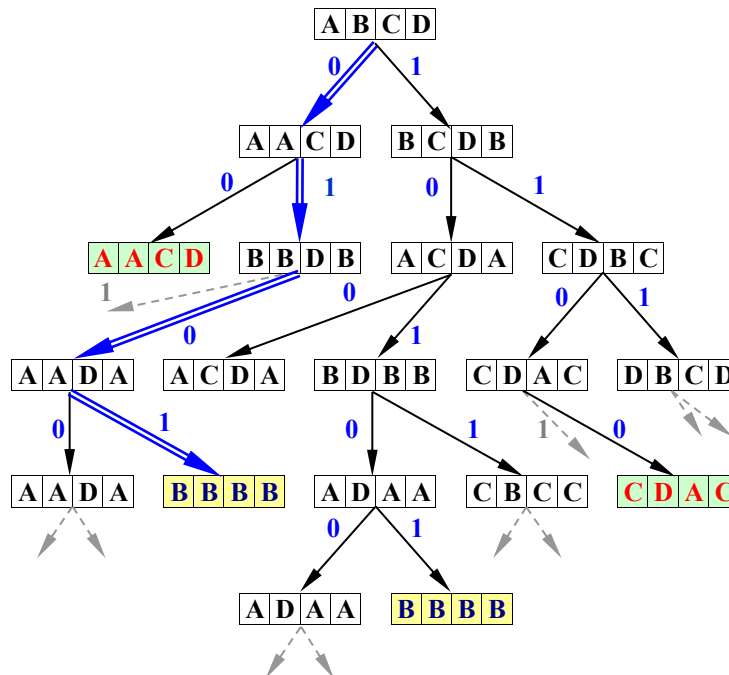
### Příklad 11.2

Sestrojte synchronizační posloupnost pro sekvenční obvod uvedený v předchozím příkladu - viz Příklad 11.1.

#### Řešení:

Úkolem je najít takové posloupnosti vstupního signálu  $x_i$ , které převedou sledovaný sekvenční obvod z jakéhokoliv vnitřního stavu do jednoho vnitřního stavu.

V tomto případě není třeba sledovat výstupní posloupnost. Zjišťujeme, jaká posloupnost vstupních proměnných uvede obvod do téhož koncového vnitřního stavu.



Obrázek 11.4: Tabulka a graf přechodů sekvenčního obvodu.

#### Výsledek:

Synchronizační posloupnost prověřovaného sekvenčního obvodu je  $x_{1,2} = 0, 1, 0, 1$  a výsledným vnitřním stavem je stav **B**.

Z příkladů lze vypožorovat, že je téměř nemožné u nijak neupravovaného sekvenčního obvodu dosáhnout stavu, kdy je možné sestavit synchronizační posloupnost pro dosažení libovolného vnitřního stavu, tedy že ne vždy je možno sestavit nastavovací posloupnost konkrétní realizace neupravovaného sekvenčního obvodu.

#### 11.1.1 Test sekvenčního obvodu

Pokud známe aktuální vnitřní stav testovaného sekvenčního obvodu, můžeme provést test sekvenčního obvodu jednou sadou testovací posloupnosti. Na správnost fungování testovaného obvodu usuzujeme podle reakce jeho výstupu.

Pro prověření činnosti sekvenčního obvodu z tohoto příkladu – viz Obrázek 11.2 – musíme na vstup přivádět takovou posloupnost vstupní proměnné, které prověří, zda testovaný obvod realizuje všechny přechody mezi vnitřními stavy. Na správnost funkce usuzujeme podle

hodnot výstupní proměnné  $z_i$ .

Po aplikaci nastavovací posloupnosti je testovaný obvod nastaven ve vnitřním stavu **B** a pro prověření jeho činnosti stačí jedna testovací posloupnost:

$$\mathbf{x}_i = 1, 0, 1, 0, 1, 0, 0, 1$$

odpovědí správně fungujícího obvodu je následující výstupní posloupnost:

$$\mathbf{z}_i = 0, 0, 1, 1, 1, 0, 0, 0$$

Pokud byl testovaný sekvenční obvod v jiném vnitřním stavu, museli bychom použít jinou posloupnost vstupní proměnné.

## 11.2 Technická řešení nastavitelnosti a pozorovatelnosti

Nemožnost zajistit kompletní nastavitelnost vnitřních stavů obecného sekvenčního obvodu a jen nepřímá možnost zajištění pozorovatelnost pomocí vektorů vstupní proměnné nutí konstruktéry vybavovat sekvenční obvody dodatečným technickým vybavením, které usnadní jak nastavování, tak zjišťování jejich vnitřního stavu.

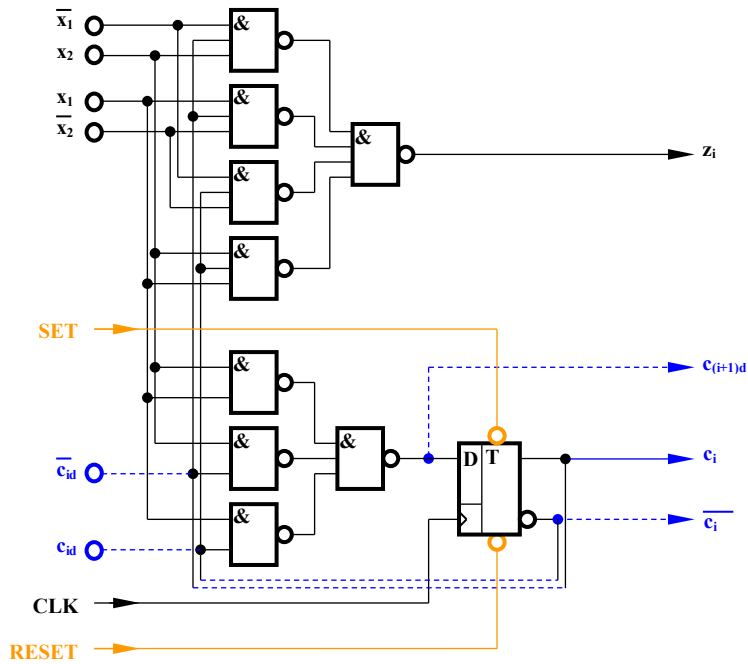
Nejjednodušším a ve všech realizacích používaným řešením nastavitelnosti počátečního vnitřního stavu je nulování paměťových elementů sekvenčního obvodu po zapnutí napájecího napětí - *POWER ON RESET*. Využívají se k tomu asynchronní vstupy klopných obvodů a u rozsáhlejších systémů programové rutiny spouštěné po - *POWER ON RESET*. Mezi další nejčastěji používané úpravy patří:

- rozpojování zpětné vazby sekvenčního obvodu,
- *Level Sensitive Scan Design* - sériový diagnostický přístup k paměti vnitřního stavu,
- *Random Access Scan* - přímý diagnostický přístup k paměti vnitřního stavu,
- *Boundary Scan* –standard IEEE 1149.1 (standard Test Access Port - TAP)

### 11.2.1 Rozpojení zpětné vazby sekvenčního obvodu

Tato úprava spočívá v možnosti odpojení výstupů klopných obvodů paměti vnitřního stavu od vstupů kombinačních obvodů generujících výstupní funkci  $\mathbf{z}_i = \mathbf{f}_z(\mathbf{x}_i, \mathbf{s}_i)$  a funkci řídicích signálů paměti následujícího stavu  $\mathbf{c}_{i+1} = \mathbf{f}_s(\mathbf{x}, \mathbf{s}_i)$ . Výsledkem této úpravy je možnost přivádět na vstupy obou kombinačních obvodů vstupní kombinace nezávisle a funkci klopných obvodů a zároveň ověřovat nastavení těchto klopných obvodů. Zapojení úpravy je znázorněno na příkladu sériové sčítačky.

Na schéma úplné sériové sčítačky - viz Obrázek 11.5 - jsou modrou barvou zvýrazněny diagnostické obvody. V tomto případě se jedná o odpojení výstupů paměti přenosu do vyššího řádu od vstupů obou kombinačních obvodů zde realizujících funkce  $\mathbf{z}_i = \mathbf{f}_z(\mathbf{x}_i, \mathbf{c}_i)$  a  $\mathbf{c}_{i+1} = \mathbf{f}_s(\mathbf{x}_i, \mathbf{c}_i)$  - což je znázorněno tečkovanou modrou čarou. Čárkovanou modrou čarou jsou zvýrazněny výstupy pro ověření správné činnosti klopného obvodu sloužícího pro uložení příznaku přenosu do vyššího řádu součtu. Plnou modrou čarou jsou znázorněny jednak vstupy dublující přímou a inverzní hodnotu signálu přenosu do vyššího řádu součtu a dále výstup paměti příznaku přenosu do vyššího řádu.



Obrázek 11.5: Schéma sériové binární sčítačky

**Poznámka:**

*Pro usnadnění aplikace sekvenčních obvodů jsou tyto vybavovány asynchronním nastavováním výchozího stavu - viz Obrázek 11.5 část zvýrazněná šedou barvou - funkcí POWER ON RESET.*

Kombinační obvody funkcí  $z_i = f_z(x_i, c_i)$  a  $c_{i+1} = f_s(x_i, c_i)$  lze po takovéto úpravě otestovat jako běžné kombinační obvody. Funkce klopného obvodu se ověří jeho nastavením a poté přečtením obou výstupů s následujícím nulováním a opětovným přečtením výstupů.

### 11.2.2 Sériový diagnostický přístup k paměti vnitřního stavu

Tato varianta zajištění nastavitelnosti a sledovanosti sekvenčního obvodu - v anglické literatuře nazývaná *Level Sensitive Scan Design* a v dalším textu bude označována zkratkou LSSD - spočívá ve specifické úpravě klopných obvodů paměti vnitřních stavů. Tyto klopné obvody jsou navrženy tak, aby umožňovaly sériový vstup a sériový výstup kódu vnitřního stavu uloženého v těchto klopných obvodech. Schéma sekvenčního obvodu je doplněno o registr počátečního vnitřního stavu, registr koncového vnitřního stavu a o obvody generování synchronizačních signálů zajišťujících vkládání a čtení vnitřního stavu testovaného sekvenčního obvodu. Blokové schéma sekvenčního obvodu s implementací technologie LSSD je uvedeno na následujícím obrázku - viz Obrázek 11.6.

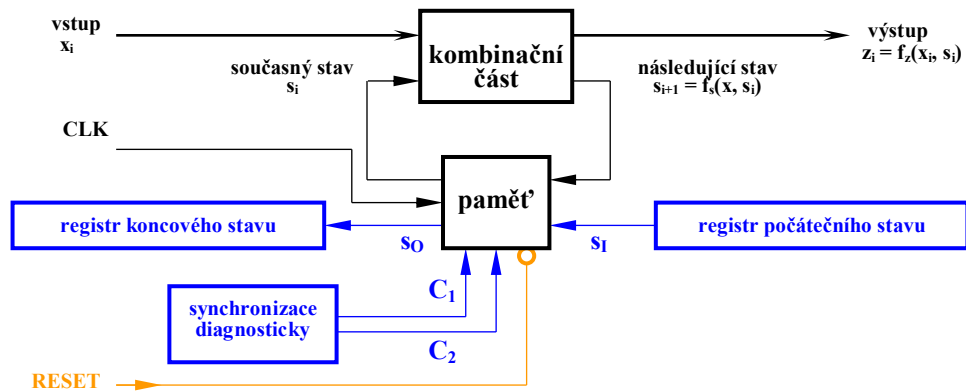
Každý klopný obvod má dva vstupy. První vstup slouží pro zápis bitu **D** následujícího vnitřního stavu sekvenčního obvodu a tedy vstupem, který je navrhován pro realizaci algoritmu sekvenčního obvodu (na schéma zvýrazněný černou barvou). Druhý vstup slouží pro vnucení bitu **S<sub>i</sub>** vnitřního stavu sekvenčního obvodu před spuštěním testovací posloupnosti.

**Poznámka:**

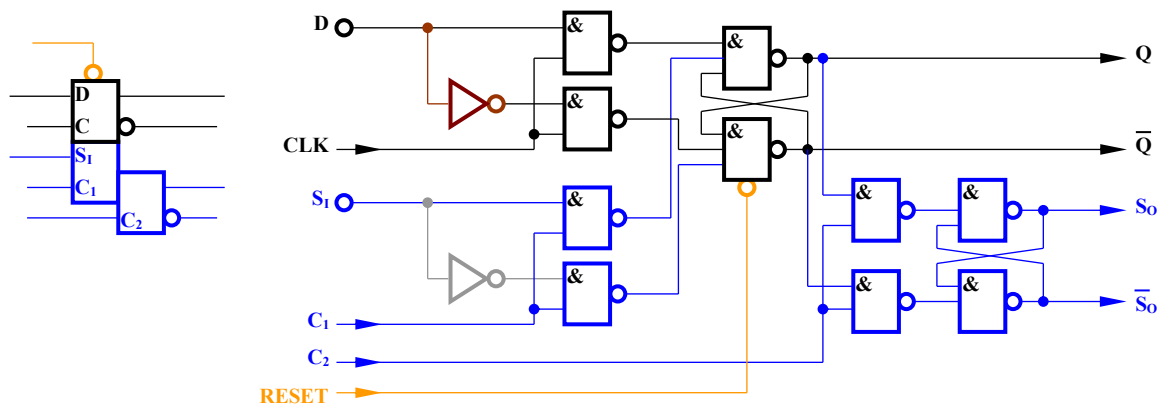
*Pro správnou funkci výše uvedeného klopného obvodu - viz Obrázek 11.7 - jsou nezbytné přímé a inverzní hodnoty zapisovaných dat. Na schéma jsou vstupní signály invertovány, ale poněvadž mohou být tyto signály k dispozici v obou verzích nebo lze použít zapojení klopného obvodu bez invertoru - viz Obrázek 11.8 - jsou inverze datového signálu zakresleny hnědou barvou.*



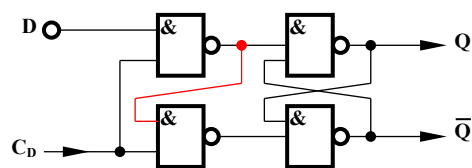
Nulování klopného obvodu paměti vnitřního stavu sekvenčního obvodu po zapnutí napájení signálem *RESET* je zakresleno barvou žlutou.



Obrázek 11.6: Blokové schéma sekvenčního obvodu s implementací technologie LSSD

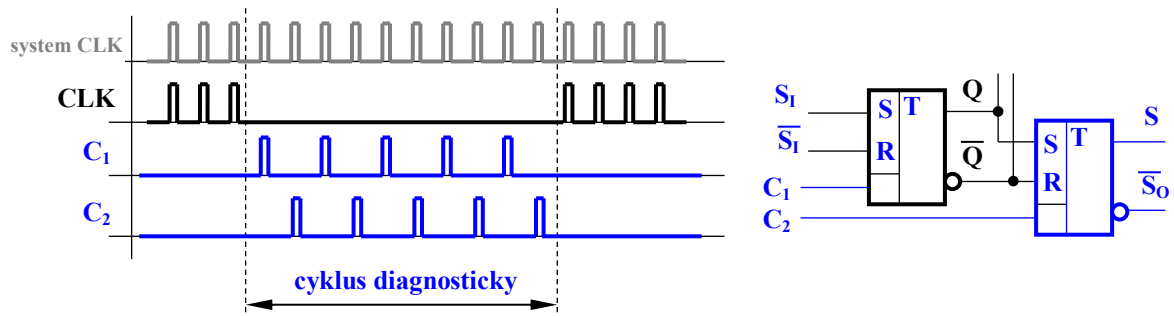


Obrázek 11.7: Symbolická značka a schéma zapojení klopného obvodu paměti vnitřního stavu technologie LSSD



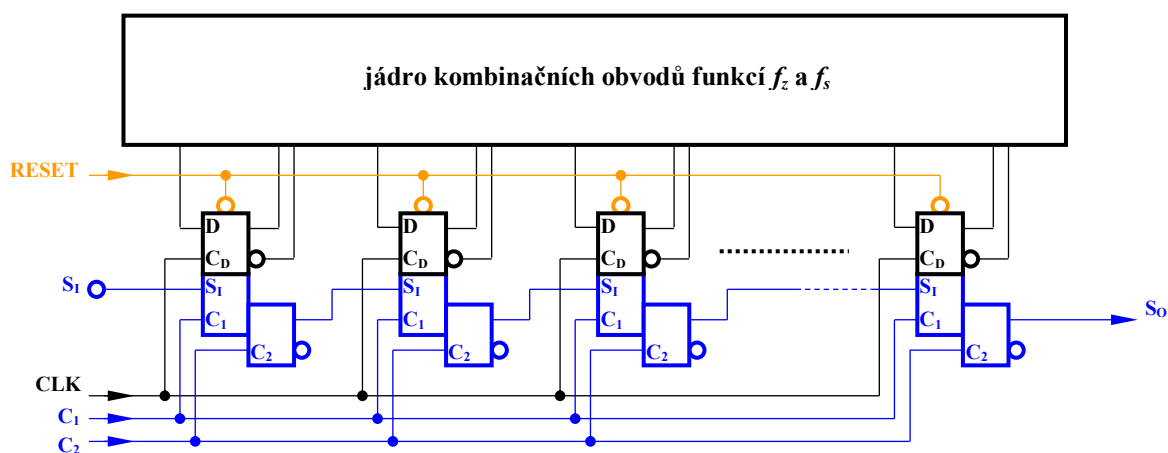
Obrázek 11.8: Schéma alternativního zapojení hladinového klopného obvodu **D**

Z hlediska konstrukce představuje tento klopný obvod paměti vnitřního stavu jeden bit registru s paralelním a sériovým zápisem a s paralelním a sériovým čtením obsahu. Použité klopné obvody mají hladinovou synchronizaci (tzv. LATCH) a tudíž pro posuv obsahu je použita dvoufázová synchronizace navzájem časově posunutými impulzy  $C_1$  a  $C_2$ . Z hlediska konstrukce je tato metoda synchronizace označována jako časování synchronizačním systémem - viz Obrázek 11.9.



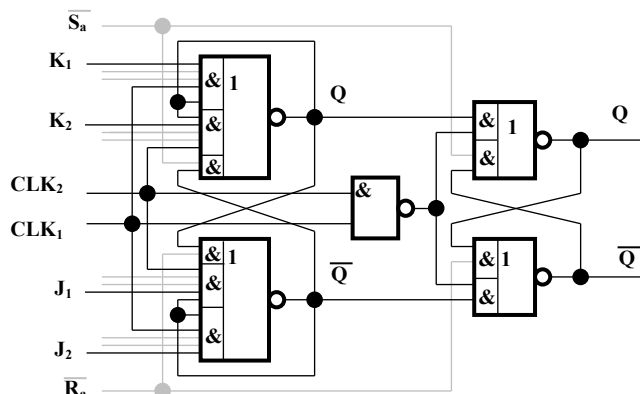
Obrázek 11.9: Princip časování klopných obvodů technologie LSSD

Struktura sekvenčního obvodu a paměti jeho vnitřního stavu s implementovanou technologií LSSD je uvedena na následujícím obrázku – viz Obrázek 11.10. Jako paměti vnitřního stavu lze použít dvouступové klopné obvody.

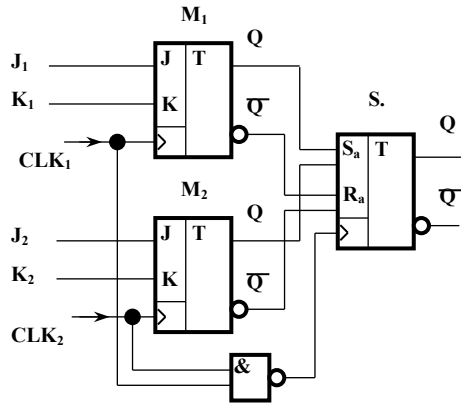


Obrázek 11.10: Struktura paměti vnitřního stavu sekvenčního obvodu s implementovanou technologií LSSD

Jako příklad lze uvést dvoufázový obvod se dvěma synchronizačními signály složený ve smyslu pravidel superpozice – viz Obrázek 11.11 a Obrázek 11.12.



Obrázek 11.11: Logické schéma dvoufázového klopného obvodu J-K se dvěma synchronizačními signály



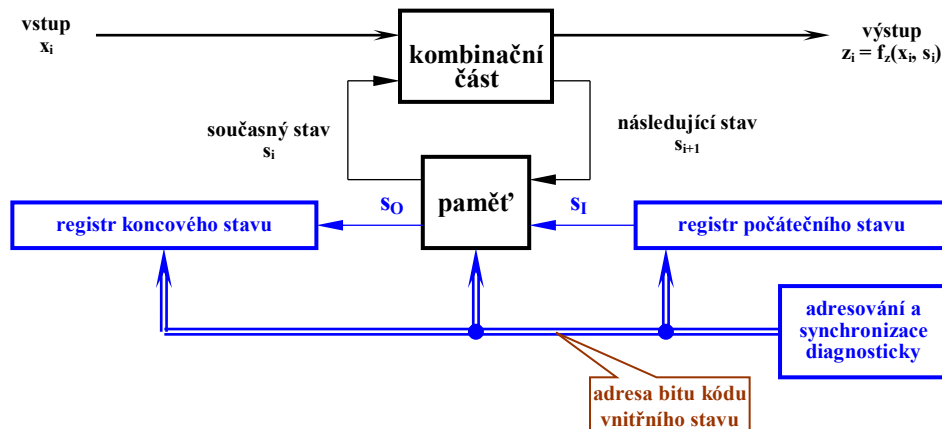
Obrázek 11.12: Logické schéma dvoufázového klopného obvodu **J-K** se dvěma samostatnými sekcemi MASTER

**Poznámka:**

*Nulování obsahu paměti vnitřního stavu sekvenčního obvodu – viz Obrázek 11.10 – po zapnutí napájení signálem RESET je zakresleno barvou žlutou.*

### 11.2.3 Přímý diagnostický přístup k paměti vnitřního stavu

Tato úprava schéma sekvenčního obvodu - v anglické literatuře nazývaná *Random Access Scan* a v dalším textu bude označována zkratkou RAS - spočívá ve v instalaci adresních dekodérů a multiplexerů pro zajištění přístupu k jednotlivým klopným obvodům paměti vnitřního stavu sekvenčního obvodu. Blokové schéma paměti sekvenčního obvodu s implementovanou technologií RAS je uvedeno níže - viz Obrázek 11.13.



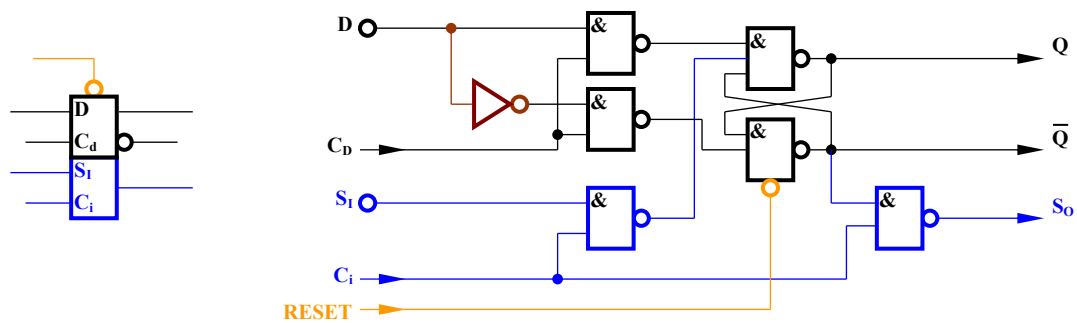
Obrázek 11.13: Blokové schéma sekvenčního obvodu s implementací technologie RAS

I v tomto případě je nezbytné kód vnitřního stavu ukládat do registrů, ale tato technologie poskytuje možnost pracovat s jednotlivými bity kódu vnitřního stavu sekvenčního obvodu a tak v některých případech zjednodušit a zrychlit nastavování a prověřování kódu vnitřního stavu během testování funkčnosti sekvenčního obvodu.

Paměť vnitřního stavu sekvenčního obvodu je složena z hladinových klopných obvodů. Ty je možné individuálně nastavovat nebo nulovat, ale z hlediska objemu technického vybavení je vhodnější využít funkce *POWER ON RESET* (ta představuje adresu "00...00") a nastavování kódu vnitřního stavu realizovat následujícím postupem ve dvou fázích:

nastavení kódu nula v paměti vnitřního stavu pomocí společného nulovacího signálu,

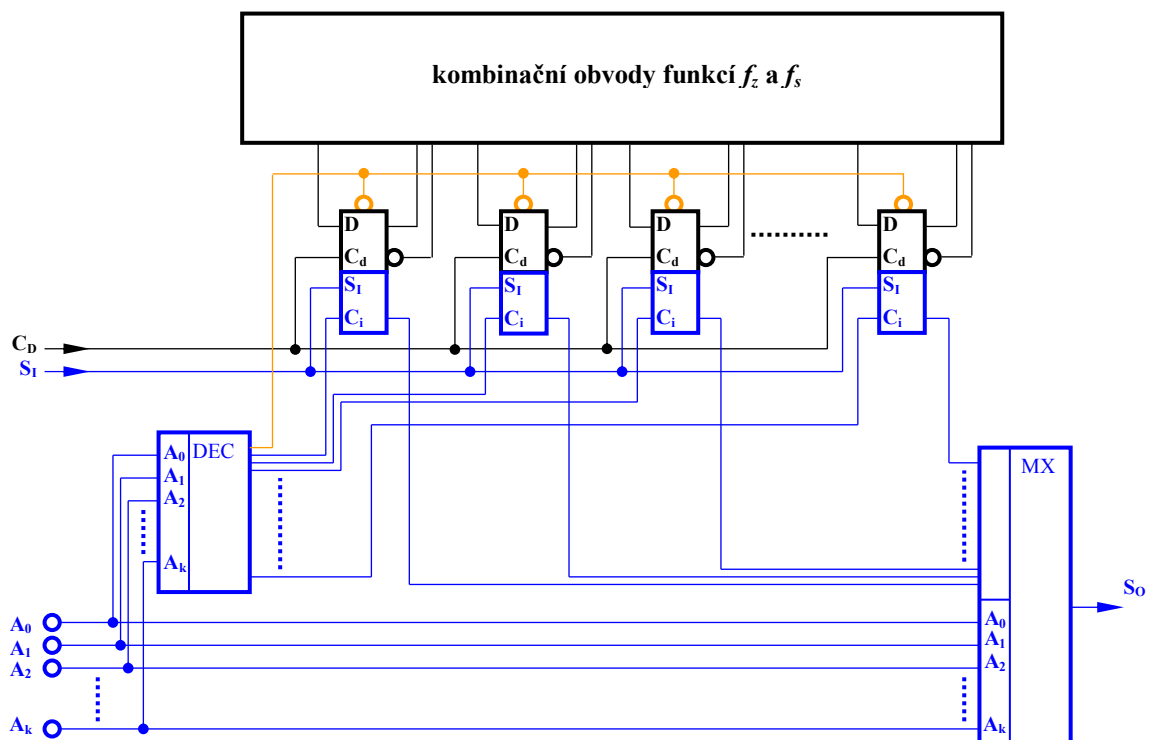
individuální nastavování "jedničkových" bitů kódu vnitřního stavu.



Obrázek 11.14: Symbolická značka a schéma zapojení klopného obvodu paměti vnitřního stavu technologie RAS

Symbolická značka a schéma zapojení klopného obvodu paměti vnitřního stavu technologie RAS je zachycena na obrázku - viz Obrázek 11.14. Zapojení klopného obvodu paměti vnitřního stavu je sice jednodušší, ale použití dekodéru a multiplexeru s sebou přináší nutnost použití značného počtu vodičů, což může v některých aplikacích přinášet značné realizační potíže.

Blokové schéma sekvenčního obvodu s implementací technologie RAS je obdobné technologii LSSD. Zapisovaný i přečtený kód vnitřního stavu sekvenčního obvodu se opět uchovávají v registrech, přičemž i v tomto případě je vhodnější použít variantu se sériovým zápisem a čtením, přičemž pozice čtení i zápisu bitu určuje adresa používaná v dekodéru a multiplexeru paměti vnitřního stavu sekvenčního obvodu.



Obrázek 11.15: Struktura paměti vnitřního stavu sekvenčního obvodu s implementovanou technologií RAS

Blok synchronizace diagnostiky u této technologie pracuje poněkud odlišně. Generuje adresy

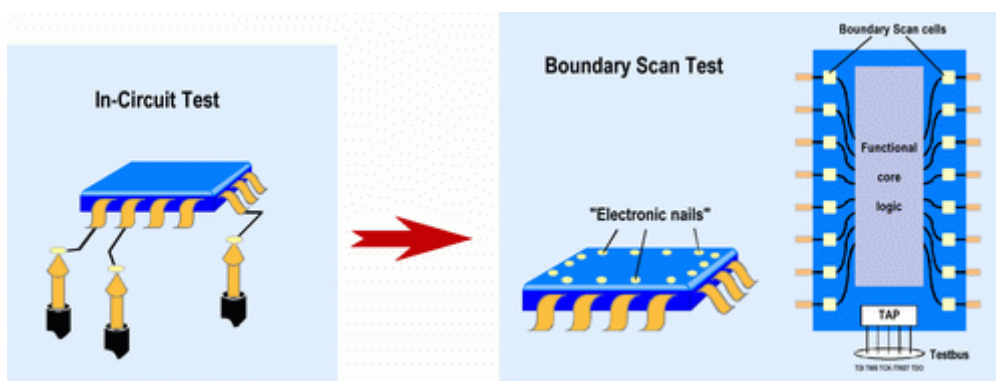
požadovaných bitů pro čtení nebo zápis a tyto jsou použity pro adresování registrů kódu vnitřního stavu a na výstupu dekodéru adres jako synchronizační signály jednotlivých klopných obvodů paměti vnitřního stavu - viz Obrázek 11.15. Čtení a zápis tedy probíhají individuálně pro každý klopný obvod a tak lze bezprostředně kontrolovat, zda se klopné obvody paměti vnitřního stavu správně nastavují nebo nulují. Počáteční nulování klopných obvodů je nutno zajistit separátně.

#### 11.2.4 Standard Boundary Scan

S rychlým vývojem zmenšování komponentů elektronických obvodů, integrovaných obvodů a desek plošných spojů (např. technologie SMT, BGA, COB a QFP) se vyvíjely i techniky a technologie jejich testování. S tím jak se vzdálenost vývodů/pinů pouzder obvodů a obvodů na deskách plošných spojů postupně zmenšovaly, začalo být mechanicky nemožné testování pomocí kontaktování hrotů a sond. Dříve byla klasickou procedurou *in-circuit* testování komponent pomocí kontaktního pole hrotů. V moderních konstrukcích již tato technika již není použitelná. Praxe si tudíž vynutila zavedení jiné metodologie.

Odpovědí na tento požadavek byl vznik standardu *Boundary-Scan*, který představuje systém testování integrovaných obvodů a desek plošných spojů programovými prostředky, umožňující rychlý a automatický způsob testování a programování integrovaných obvodů a elektronických výrobků. Skupina *Joint Test Action Group* zpracovala návrh standardu, který byl v roce 1990 schválen jako norma IEEE 1149.1, tzv. Standard Test Access Port (TAP). V této normě je standardizováno taktéž rozhraní JTAG. Jedná se nízkonákladový a v současné době jednoduchý, automatický testovací systém, který lze snadno instalovat na běžných PC.

Standard *Boundary-Scan* poskytuje možnost testovat jak integrované obvody, desky plošných spojů, paměti Flash apod., které tento standard podporují, tak i prvky, které nezbytné technické vybavení implantováno nemají. Nástroje programového vybavení standardu umožňuje přehledně lokalizovat a zobrazit zjištěné chyby.

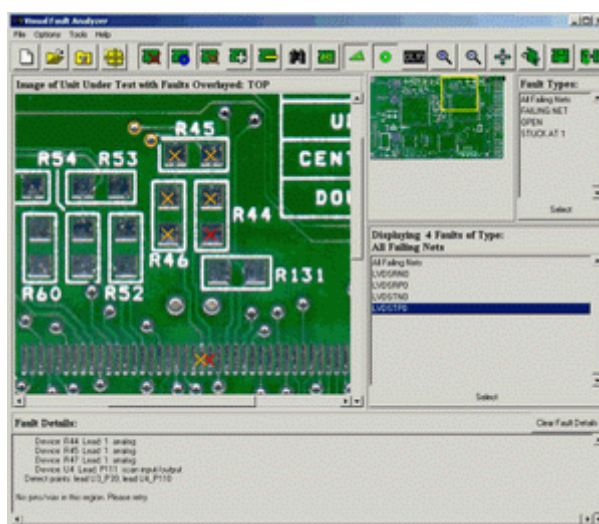


Obrázek 11.16: Znárodnění principu testování obvodů metodou *in-circuit test* s použitím jehlového pole a metody boundary-scan

Standard *Boundary-Scan* je tvořen rozhraním JTAG, programovým vybavením a vývojovými nástroji pro sestavování a vyhodnocování testů elektronických zařízení. Kompaktní testovací systém *Boundary-Scan* představuje techniku umožňující programově testovat a kontrolovat správnost realizace a funkci integrovaného obvodu, desky plošných spojů i celých systémů a zařízení bez nutnosti instalovat měřící body a fyzicky připojovat měřící hroty nebo sondy na vývody pouzder jednotlivých obvodů. Navíc vývojové nástroje standardu poskytují techniky pro testování i těch obvodů a částí desek plošných spojů, které přímo standard nepodporují.

Systém *Boundary-Scan* je metodika umožňující dodržet zásady *nastavitelnosti* testované jednotky - tedy *controllability* - a její *pozorovatelnosti* - *observability* – prostřednictvím programového vybavení. Princip implementace tohoto systému spočívá v přidání pomocných obvodů - *hraničních buněk* - k vývodům integrovaného obvodu tak, že oddělují jeho logické jádro od fyzických vývodů a umožňují číst obsah nebo vnučovat logické hodnoty na tyto vývody. Tento systém je komerčně dostupný v řadě variant a je k dispozici se běžně používá pro následující úkony:

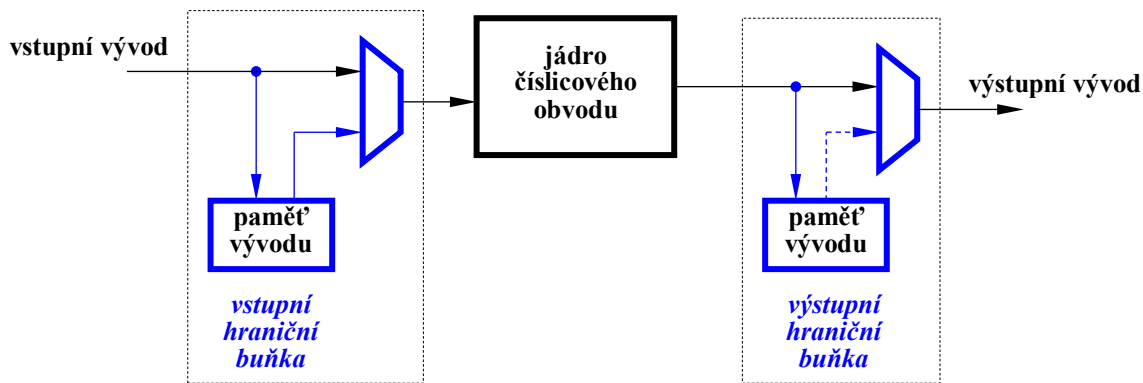
- ladění prototypu infrastruktury hardwaru od montážních chyb a zkontrolovat funkčnost logických bloků,
- testování integrity propojení paměťových bloků od zkratů a přerušených spojení,
- programování CPLD a konfiguraci EPROM paměti obvodů FPGA všech výrobců, přes signálové rozhraní nebo TAP (Test Access Port),
- testování pomocí konektorů. Přídavné hardwarové moduly DIOS (Digital I/O Scan = digitální skenování vstupů/výstupů) vytváří možnost jednoduše rozšířit testování až na hlavní svorkovnice na kraji desek nebo na libovolné testovací body,
- *boundary-scan* řadič vybavený interní pamětí může jednoduše programovat Flash paměti přenosovými rychlostmi až 2 MB/min



Obrázek 11.17: Příklad grafického zobrazení chyb na desce plošných spojů

#### 11.2.4.1 Princip činnosti metody *Boundary-Scan*

Jak již bylo uvedeno v předchozím textu, principem implementace systému *Boundary-Scan* spočívá v přidání pomocných obvodů - *hraničních buněk* - k vývodům integrovaného obvodu tak, že oddělují jeho logické jádro od fyzických vývodů a umožňují číst obsah nebo vnučovat logické hodnoty na tyto vývody. Blokové schéma implementace hraničních buněk je znázorněno na následujícím obrázku - viz Obrázek 11.18. Takto lze číst obsah všech vývodů integrovaného obvodu. Pro úplnou implementaci systému *Boundary-Scan* je třeba umožnit vnučování vstupních testovacích vektorů na vstupy jádra integrovaného obvodu.



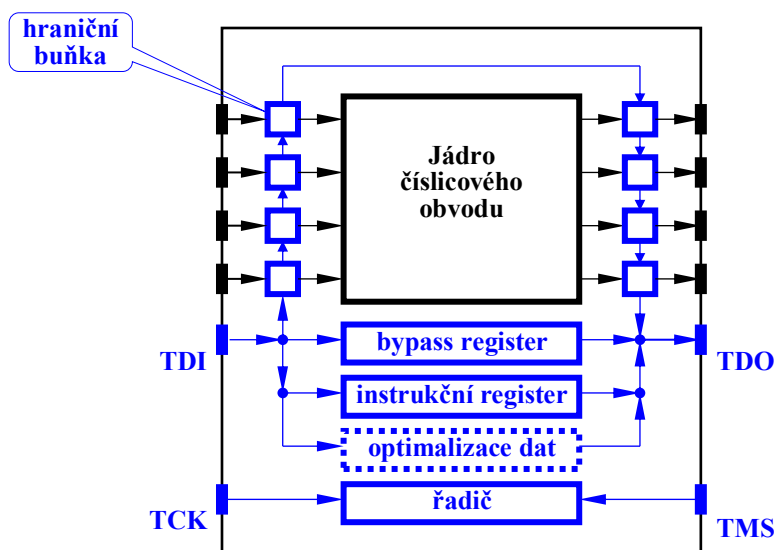
Obrázek 11.18: Principiální schéma implementace obvodů *Boundary-Scan* do logiky číslicového obvodu

Činnost vstupních a výstupních hraničních buněk je řízena řadičem ovládaným čtyřmi nebo pěti vodiči TAP (Test Access Port). Řadič TAP je konečný automat se šestnácti stavy, který řídí činnost hraničních buněk číslicového obvodu. Ovládací signály řadiče TAP jsou přiváděny na stejnojmenné vývody - viz níže:

- Test Clock (**TCK**),
- Test Mode Select (**TMS**),
- Test Data In (**TDI**),
- Test Data Out (**TDO**),
- Test Reset (**TRST**) - asynchronní nulování řadiče TAP. Není použit u všech realizací.

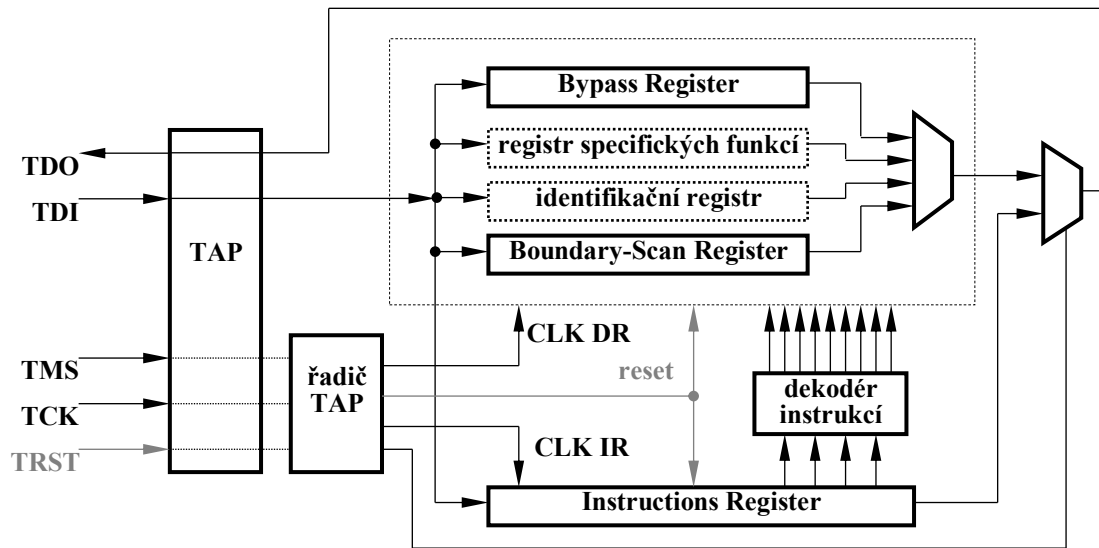
Signály **TCK** a **TMS** řídí posloupnost stavů řadiče TAP řadiče. Vývody **TDI** a **TDO** zprostředkovávají tok dat testovacích vektorů a odezev přes všechny zřetěžené obvody testované metodou **B-ST**. Režim činnosti **B-ST** je představuje množinu činností:

- **EXTEST** - testování okolí obvodu,
- **INTEST** - testování vnitřní funkce obvodu,
- **RUNBIST** - test s autonomním generováním testovacích vektorů a kompresí odezev.



Obrázek 11.19: Schéma začlenění systému *Boundary-Scan* do pouzdra integrovaného obvodu

Princip začlenění systému *Boundary-Scan* do pouzdra integrovaného obvodu a blokové schéma obvodů systému *Boundary-Scan* jsou zobrazeny na obrázcích - Obrázek 11.19 a Obrázek 11.20.



Obrázek 11.20: blokové schéma obvodů systému *Boundary-Scan*

Bypass Register (**BR**) umožňuje přenos dat ze vstupu **TDI** na výstup **TDO** prostřednictvím jediného klopného obvodu. Použití instrukce **BYPASS** umožňuje test jednotlivých obvodů zapojených v kaskádě prostřednictvím vývodů **TDI** a **TDO**. Takto mohou být spojovány i větší celky - například celé desky elektroniky. V tomto případě je pro všechny v tomto kroku netestované obvody použita instrukce **BYPASS**.

Nejdůležitějším registrem systému je registr *Boundary-Scan Register* (**BSR**), který je tvořen klopnými obvody hraničních buněk. Ty jsou připojeny k vývodům integrovaného obvodu. Tento registr slouží pro testování správnosti připojení integrovaného obvodu na plošný spoj a může být zdrojem testovacích vektorů i záchytným registrem odezev obvodu na prováděný test.

*Identifikační registr* (**IR**) je registr s pevným obsahem, který je nutný pro rozpoznání typu obvodu. Obsah registru **IR** je standardizován. V registru **IR** je obsah, který je porovnáván s technickými podmínkami a slouží k rychlému rozpoznání typu obvodu a typu jeho personifikace zákaznickým obsahem.

*Registr specifických funkcí* je registr nebo skupina registrů, pracující jako pseudonáhodné generátory testovacích vektorů. Mohou sloužit jako zdroje pseudonáhodně generovaných testovacích vektorů přiváděných na vstupy testované jednotky. Mohou však sloužit také jako příznakové analyzátoři a provádět kompresi odezev testovaného obvodu.

Další rozvoj metody *Boundary-Scan* pokračuje rozvojem a standardizací testování analogových obvodů. Řízení testů je prováděno tak, aby testy analogových obvodů mohly být zařazeny do metodiky smíšených obvodů AMS (Analog Mixed Signal). Standard pro testování smíšených a analogových obvodů má označení IEEE 1149.5.

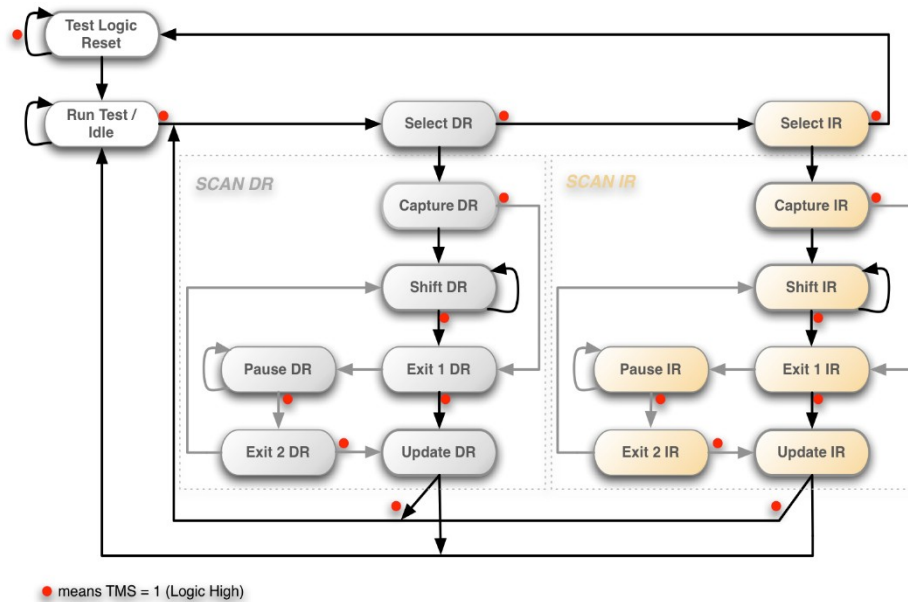
Podpora metody B-ST pomocí návrhového prostředí jazyka VHDL (VHSIC Hardware Description Language) je zajištěna standardizací tzv. BSDL (Boundary-Scan Design



Language). Jazykem BSDL jsou popsána obvody podporující testování metodou B-ST, přičemž tyto obvody mohou být simulovány i bez implementace podpory v aplikaci.

#### 11.2.4.2 Řadič Test Access Port (TAP)

Tento řadič je řízen signálem přiváděným na vstup TMS. Je realizován jako 16 stavový konečný automat. Řadič TAP řídí činnost celého systému JTAG.



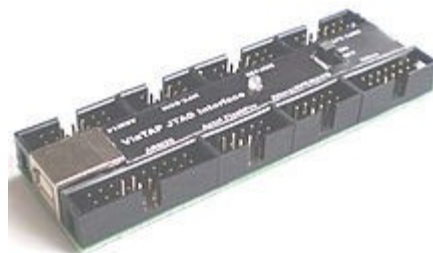
Obrázek 11.21: graf přechodů řadiče TAP

Všechny stavy stavového automatu TAP mají dva výstupy. Všechny přechody mezi vnitřními stavy jsou řízeny signálem TMS, který je vzorkován signálem TCK přiváděným na stejnojmenný vývod.

V systému jsou implementovány dvě cesty pro zápis nebo čtení dat - datové registry a instrukční registr. Datové registry jsou především *Bypass Register (BR)*, *ID Code Register (IDCODES)* a *Boundary-Scan Register (BSR)*. Instrukční registr je v systému implementován jeden, a jak naznačuje název, je určen k ukládání kódů instrukcí systému *Boundary-Scan*.

Podrobný popis stavového automatu řadiče TAP je uveden ve standardu IEEE 1149.1 popisujícím podrobně strukturu a funkci řadiče TAP.

#### 11.2.4.3 JTAG konektory



Obrázek 11.22: Příklad realizace konektoru JTAG

Přestože JTAG je standardizovaný, konektory na připojení JTAG adaptéru již nikoliv. Většina výrobců používá vlastní rozmístění vývodů a špiček konektoru, přičemž je většinou použit

"pin header" s roztečí 2.54mm. Obecně se mezi výrobci rozšířilo jen pár zapojení, zejména MIPS EJTAG, ARM JTAG atd.

#### 11.2.4.4 Instrukce systému Boundary-Scan

Instrukce jsou definovány standardem IEEE 1149.1. Pokud je jejich obsah a struktura akceptována, je zajištěna kompatibilita pro široké spektrum elektronických součástek:

- **BYPASS** - tato instrukce umožní přímé propojení vývodů **TDI** a **TDO** přes jednobitový *Bypass Register (BR)*. Tato instrukce umožňuje obejít adresovaný obvod a tak urychlit testování dalšího nebo dalších obvodů v tzv. řetězci JTAG,
- **EXTEST** - instrukce připojuje *Boundary Scan Register (BSR)* k vývodům **TDI** a **TDO**. Logické hodnoty na všech vývodech pouzdra integrovaného obvodu jsou zapsány do paměti vývodu - viz stav *Capture DR* řadiče JTAG. Poté jsou nové logické hodnoty stavem *Shift DR* řadiče JTAG posunuty do **BSR**. Aktualizované hodnoty vývodů jsou poté ke vstupům logického obvodu připojeny stavem *Update DR* řadiče JTAG,
- **SAMPLE/PRELOAD** - tato instrukce opět propojuje **BSR** s vývody **TDI** a **TDO**. Testované zařízení zůstává ve standardním funkčním stavu. Během této instrukce je **BSR** přístupný pro čtení svého obsahu a tak umožňuje průběžné snímání vstupních a výstupních úrovní testovaného obvodu bez ovlivňování jeho funkce. Instrukce také slouží k nastavení testovacích úrovní do **BSR** před spuštěním instrukce **EXTEST**,

Další běžně dostupné instrukce jsou následující:

- **IDCODE** - tato instrukce připojuje registr **IDCODE** k vývodům **TDI** a **TDO**. Umožňuje tak nastavovat nebo číst identifikační kód obvodu a tím umožnit testovacímu programu vybrat vhodný podprogram a testovací data pro ověření průchodnosti tohoto integrovaného obvodu. Identifikační kód a číslo revize integrovaného obvodu umožní výběr jeho popisu souboru v *Boundary-Scan Design Language (BSD)*. Soubor obsahuje podrobné informace o konfiguraci *Boundary-Scan* systému pro daný typ integrovaného obvodu.
- **INTEST** - opět připojuje *Boundary Scan Register (BSR)* k vývodům **TDI** a **TDO**. Zatímco instrukce **EXTEST** umožňuje nastavit a číst logické hodnoty na vývodech integrovaného obvodu, instrukce **INTEST** pracuje s vnitřními paměťovými elementy testovaného obvodu.

Podrobný a úplný popis instrukcí *Boundary-Scan* je uveden ve standardu IEEE 1149.1 popisujícím podrobně strukturu a funkci JTAG.

#### 11.2.4.5 Testování integrovaných obvodů

Jednotlivé hraniční buňky systému *Boundary-Scan* (*Boundary Scan Cells - BSCs*) jsou vzájemně propojené a řízené TAP řadičem implementovaným na čipu daného integrovaného obvodu společně s funkční logikou (JTAG logikou). To umožňuje aplikovat sériové nasouvání externích testovacích vektorů simulujících hodnoty vstupních a následně vysouvat výsledné hodnoty z výstupních pinů. Kaskádní zapojení hraničních buněk začíná vstupem **TDI** (*Test Data Input*) a je ukončena výstupem **TDO** (*Test Data Output*).

Standard IEEE 1149.1 definuje jednotné provedení interní řídicí logiky i rozhraní komunikační sběrnice. Ta provádí řídicí spojení mezi nadřazeným testovacím zařízením (většinou standardní PC) a testovanou součástkou. Použití této architektury umožňuje zjišťovat a lokalizovat externí i interní výrobní chyby jako jsou zkratovaná nebo naopak

rozpojená propojení či aplikační chyby špatně vzájemně propojených obvodů a jejich programového vybavení (např. FPGA).

Toho je dosaženo automatizovaným generátorem testovacích vzorků ATPG (*Automatic Test-Pattern Generation*) nebo automatickým testováním BIST (*Built-In Self-Test*). V současné době se k tomu navíc přidala možnost *in-circuit* programování CPLD, FPGA a MCU, což celkově zjednodušuje náklady a rychlost vývoje zařízení.

Standardizace řadiče TAP *Boundary-Scan* architektury byla široce akceptována většinou velkých hardwarových i softwarových výrobců součástek, testerů i testovacích softwarů a vývojových prostředků, jako jsou CAE (*Computer-Automated Engineering tool*). Jde tedy o univerzální testovací prostředek nezávislý na platformě, podporující všechny typy vyvíjených desek plošných spojů od různých výrobců a tedy pro testování různých desek lze použít různé libovolné programové vybavení podporující tento systém.

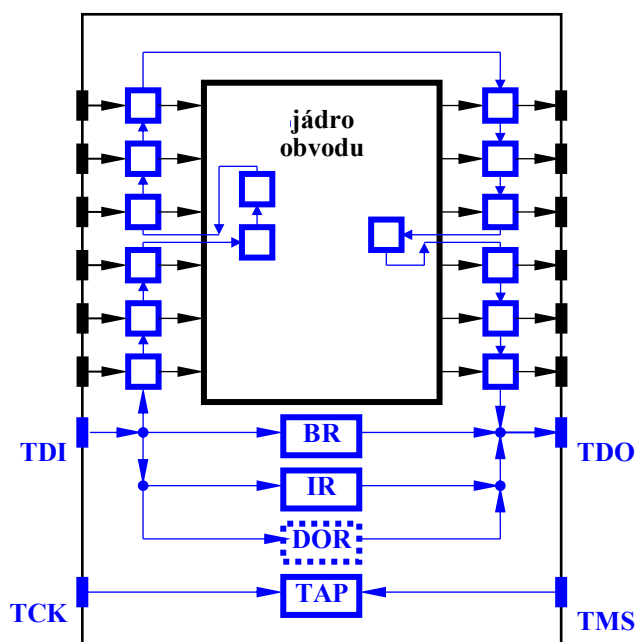
V současné době je již velmi široká nabídka integrovaných obvodů podporující JTAG a *Boundary-Scan* architekturu, takže prakticky lze najít vždy součástku potřebného typu, která tuto architekturu podporuje. Navíc již dnes testovací software umožňuje provádět testování i těch obvodů a částí desek, které *Boundary-scan* nepodporují. Mezi ně například patří jednoduché logické prvky (NAND, XOR apod.), klopné obvody, registry, paměťové čipy RAM, FLASH nebo ROM, různé paralelní komunikační rozhraní, řadiče apod. Tyto části jsou totiž zahrnuty do bloků označené jako cluster, které jsou pak testovány prostřednictvím na ně napojených obvodů podporujících *Boundary-Scan*. Jejich prostřednictvím lze na jejich výstupech generovat testovací úrovně, na které cluster bloky reagují normálně dle své funkce. Pokud některé části nejdou testovat nebo je nechceme testovat, je možno je nahradit emulací testovacím softwarem - podle jejich podstaty zkratem nebo rozpojeným spojením. Někteří výrobci nabízejí vestavěné generátory testovacích signálů paralelních vstupů a výstupů, které umožňují vývojářům desek implementovat automatické testování BIST (*Built-In Self-Test*) nejen na úrovni součástek, ale i celých desek, které poskytují vysokou kontrolu chyb při generování velmi malého počtu testovacích vzorků signálů.

#### 11.2.4.6 Interní testování integrovaných obvodů

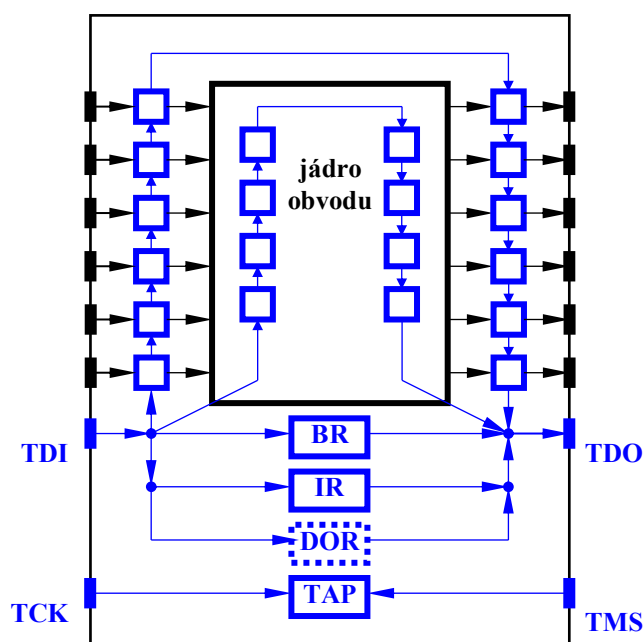
Doposud byla diskutována možnost testování integrovaných obvodů snímáním logických úrovní na vstupních a výstupních špičkách integrovaných obvodů. Systém *Boundary-Scan* však po určitých úpravách umožňuje nastavovat a číst hodnoty logických úrovní uvnitř struktury jádra integrovaného obvodu. Pro zajištění této funkce je nutno jádro vybavit paměťovými diagnostickými obvody umožňujícími nastavovat a fixovat logické hodnoty v místech jádra integrovaného obvodu, která jsou obtížně nebo nenastavitelná a nesledovatelná ze vstupů a výstupů obvodu.

Jedna z možností je začlenit vnitřní diagnostické paměťové buňky do řetězce **BSR**. Principiální schéma začlenění této funkcionality do struktury integrovaného obvodu je uvedeno na níže - viz Obrázek 11.23.

Druhou možností je pro vnitřní diagnostické paměťové buňky implementovat vlastní sekvenční řetězec - Obrázek 11.24 - a ten potom ovládat vlastními instrukcemi. K tomuto řešení se přiklonila řada výrobců a je vedeno úsilí o standardizaci tohoto řešení.



Obrázek 11.23: Schéma začlenění systému *Boundary-Scan* do pouzdra integrovaného obvodu



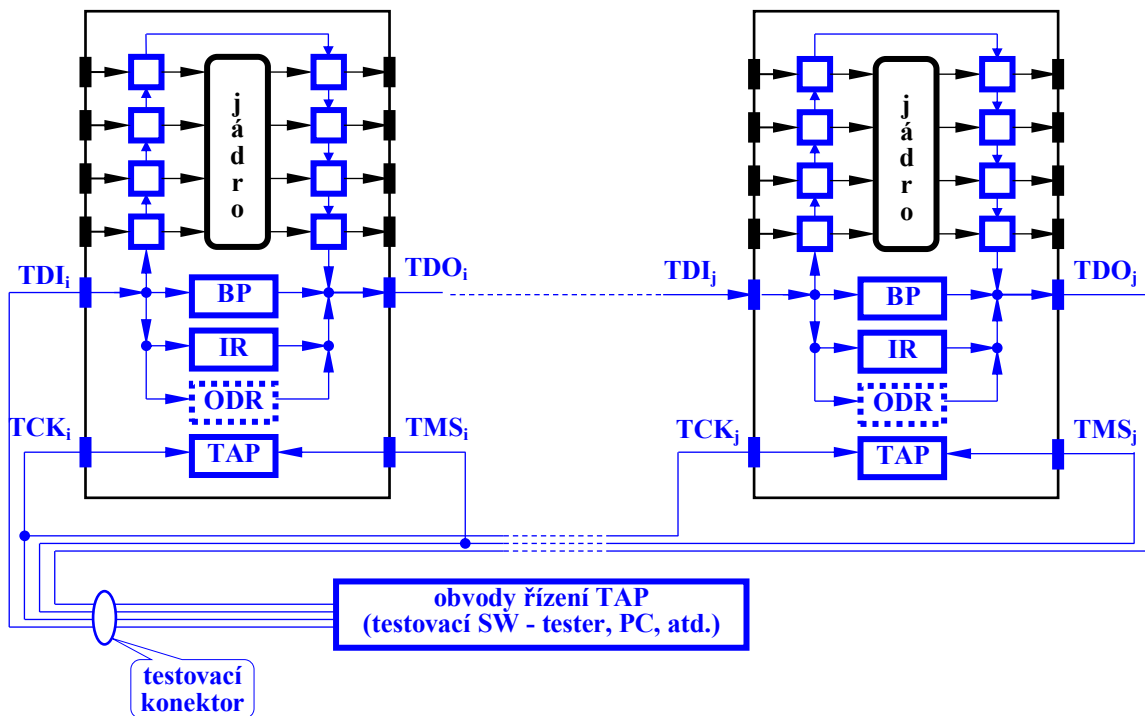
Obrázek 11.24: Schéma začlenění systému *Boundary-Scan* do jádra integrovaného obvodu

Tato metoda přímo souvisí s implementací požadavku dodržet zásady *nastavitelnosti* testované jednotky - tedy *controllability* - a *pozorovatelnosti* - *observability*. Umožňuje jednoduchým způsobem - konkrétně zde specifickými instrukcemi systému - od sebe oddělit testování metodou *black box* a testování *in-build*, která předpokládá využívat možnosti přímého zápisu a čtení dat z vnitřních registrů testovaného sekvenčního obvodu. Struktura a vlastnosti řady konkrétních realizací sekvenčních obvodů neumožňují dostatečně pružným způsobem nastavovat a detekovat jejich vnitřní stavy.

### 11.2.4.7 Testování na úrovni desek

Systém *Boundary-Scan* umožňuje testování na úrovni celých desek nebo dokonce celých sestav desek vložených do konektorového pole.

Kompletní systém vyžaduje určitou úpravu popsanou ve standardu. Pro testování desky plošných spojů jsou vývody **TDI** a **TDO** jednotlivých obvodů zapojeny do série - řetězce. Všechny TAP signály je nutno kontrolovat na integritu. Pro zajištění správného časování lze v celém řetězci aplikovat "bufferování" signálů **TCK** a **TMS** a tím zajistit správné časování i při použití maximálních rychlostí práce celého systému - viz Obrázek 11.25.



Obrázek 11.25: Schéma Propojení jednotlivých *Boundary-Scan* obvodů do jediného testovacího řetězce

U desek zasunutých do konektorového pole lze vhodnou úpravou implementovat paralelní zapojení řetězců pro každou desku zvlášť.

## Seznam obrázků

Obrázek 11.1: Blokové schéma sekvenčního obvodu .....	2
Obrázek 11.2: Tabulka a graf přechodů sekvenčního obvodu.....	4
Obrázek 11.3: Rozlišovací posloupnost sekvenčního obvodu.....	5
Obrázek 11.4: Tabulka a graf přechodů sekvenčního obvodu.....	6
Obrázek 11.5: Schéma sériové binární sčítačky .....	8
Obrázek 11.6: Blokové schéma sekvenčního obvodu s implementací technologie LSSD .....	9
Obrázek 11.7: Symbolická značka a schéma zapojení klopného obvodu paměti vnitřního stavu technologie LSSD .....	9
Obrázek 11.8: Schéma alternativního zapojení hladinového klopného obvodu <b>D</b> .....	9
Obrázek 11.9: Princip časování klopných obvodů technologie LSSD.....	10
Obrázek 11.10: Struktura paměti vnitřního stavu sekvenčního obvodu s implementovanou technologií LSSD .....	10
Obrázek 11.11: Logické schéma dvoufázového klopného obvodu <b>J-K</b> se dvěma synchronizačními signály.....	10
Obrázek 11.12: Logické schéma dvoufázového klopného obvodu <b>J-K</b> se dvěma samostatnými sekcemi MASTER.....	11
Obrázek 11.13: Blokové schéma sekvenčního obvodu s implementací technologie RAS .....	11
Obrázek 11.14: Symbolická značka a schéma zapojení klopného obvodu paměti vnitřního stavu technologie RAS.....	12
Obrázek 11.15: Struktura paměti vnitřního stavu sekvenčního obvodu s implementovanou technologií RAS .....	12
Obrázek 11.16: Znázornění principu testování obvodů metodou <i>in-circuit test</i> s použitím jehlového pole a metody boundary-scan.....	13
Obrázek 11.17: Příklad grafického zobrazení chyb na desce plošných spojů .....	14
Obrázek 11.18: Principiální schéma implementace obvodů <i>Boundary-Scan</i> do logiky číslicového obvodu .....	15
Obrázek 11.19: Schéma začlenění systému <i>Boundary-Scan</i> do pouzdra integrovaného obvodu .....	15
Obrázek 11.20: blokové schéma obvodů systému <i>Boundary-Scan</i> .....	16
Obrázek 11.21: graf přechodů radiče TAP .....	17
Obrázek 11.22: Příklad realizace konektoru JTAG.....	17
Obrázek 11.23: Schéma začlenění systému <i>Boundary-Scan</i> do pouzdra integrovaného obvodu .....	20
Obrázek 11.24: Schéma začlenění systému <i>Boundary-Scan</i> do jádra integrovaného obvodu ..	20
Obrázek 11.25: Schéma Propojení jednotlivých <i>Boundary-Scan</i> obvodů do jediného testovacího řetězce.....	21

## Literatura

- Hlavička J.: Diagnostika a spolehlivost, Vydavatelství ČVUT, Praha (1990), ISBN 80-01-01846-6
- Musil, V., Vlček, K.: Diagnostika elektronických obvodů, TEMPUS Equator S\_JEP-09468-95, ÚMEL, FEI VUT v Brně (1998)
- Hlavička, J., Kottek, E., Zelený, J.: Diagnostika Elektronických číslicových obvodů, Praha SNTL (1982)
- Drábek, V.: Spolehlivost a diagnostika, VUT Brno, (1983)
- Hławiczka, A.: P1149, Warszawa (1993), ISBN 83-204-1518-7
- Poupa, J., Pinker, M.: Číslicové systémy a jazyk VHDL, BEN Praha (2006), ISBN 80-7300-198-5
- Vlček, K.: Komprese a kódová zabezpečení v multimediálních komunikacích, BEN Praha (2000, 2004), ISBN 80-86056-68-6, ISBN 80-7300-134-9
- Vlček, K.: Návrh obvodů pomocí VHDL, Sdělovací technika, 1998, č. 2, str. 20