

Nástroje pro programování ARM bez použití OS

Možnosti programování pro různé varianty opračního systému (C/C++, ASM)

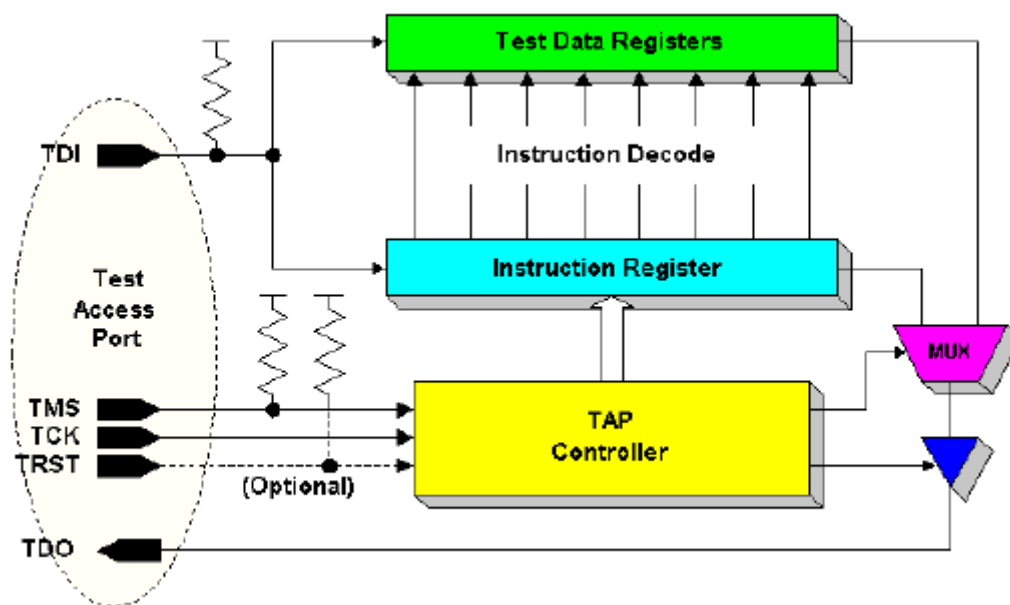
- Win CE
 - Microsoft Visual Studio s příslušnou SDK na počítači PC
- Linux
 - Cross compiler na bázi GCC pod OS Linux na počítači PC
- Android
 - SDK, Java, Eclipse na PC
 - NDK
- Bez OS
 - Cross compiler na bázi GCC pod OS Linux na počítači PC
 - **Yagarto tools, Eclipse, Debug server na počítači PC**

Možnosti testování a ladění programů:

- Připojení USB, RS232, DNW
 - nemožnost ladění na úrovni zdroj. Textu
 - diagnostika kooperací testovaného programu (RS232)
 - ovládání běhu programu přes RS232
- Připojení přes JTAG
 - Plnohodnotné ladění na úrovni zdrojového kódu
 - kanály USB, RS232 mohou být využity programem jinak
 - potřeba speciálního interface JTAG

Popis rozhraní JTAG

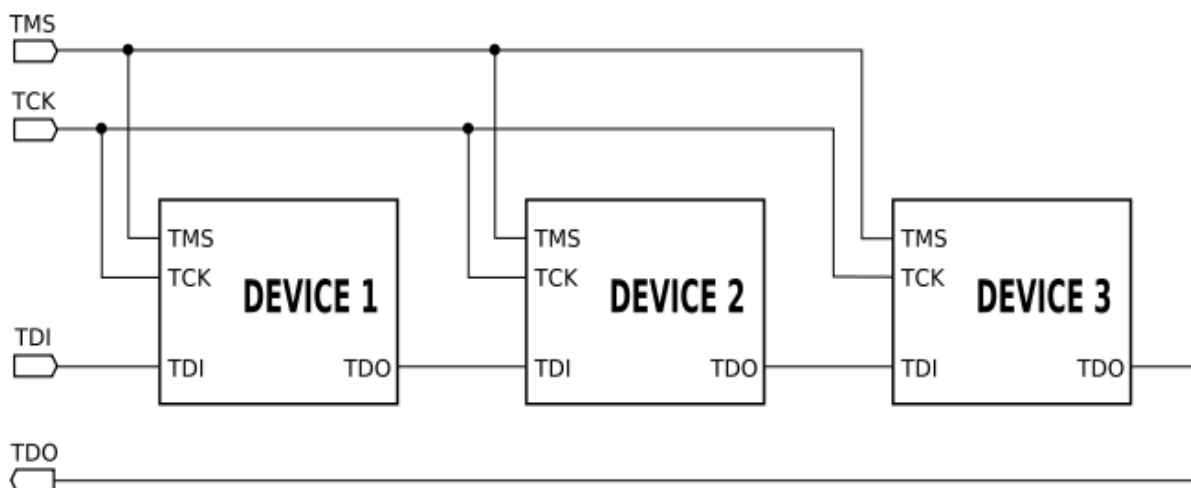
- Test Access Port a architektura Boundary-Scan podle normy IEEE 1149.1 1990
- Skupina výrobců integrovaných obvodů a jejich prodejců označovaná jako JTAG (Joint Test Actoin Group) usiluje o dosažení dohody, která by stanovila princip konstrukce integrovaných obvodů.
- Cílem této skupiny je, aby metodika testování integrovaných obvodů byla použitelná i v případě, že se na jedné desce vyskytují komponenty od různých výrobců.



- Současná verze této metodiky, reprezentovaná normou IEEE 1149.1, je výsledkem sjednocení názorů výrobců integrovaných obvodů a jejich zákazníků. Je to kompromis mezi snahou o respektování požadavků diagnostiky a snahou o co nejnížší výrobní náklady.
- Doporučení JTAG není striktní v tom smyslu, že by přesně určovalo, jak testovací obvody (Boundary-Scan buňky) konstruovat. Zaměřuje se spíše na definování rozhraní integrovaného obvodu a způsob komunikace na tomto rozhraní.
- Norma IEEE 1149.1 tedy definuje testovací logické obvody, které mohou být zařazeny do integrovaných obvodů za účelem testování propojení mezi jednotlivými komponentami na již osazených deskách, testování integrovaného

obvodu samotného a **sledování chování integrovaného obvodu při jeho normální činnosti.**

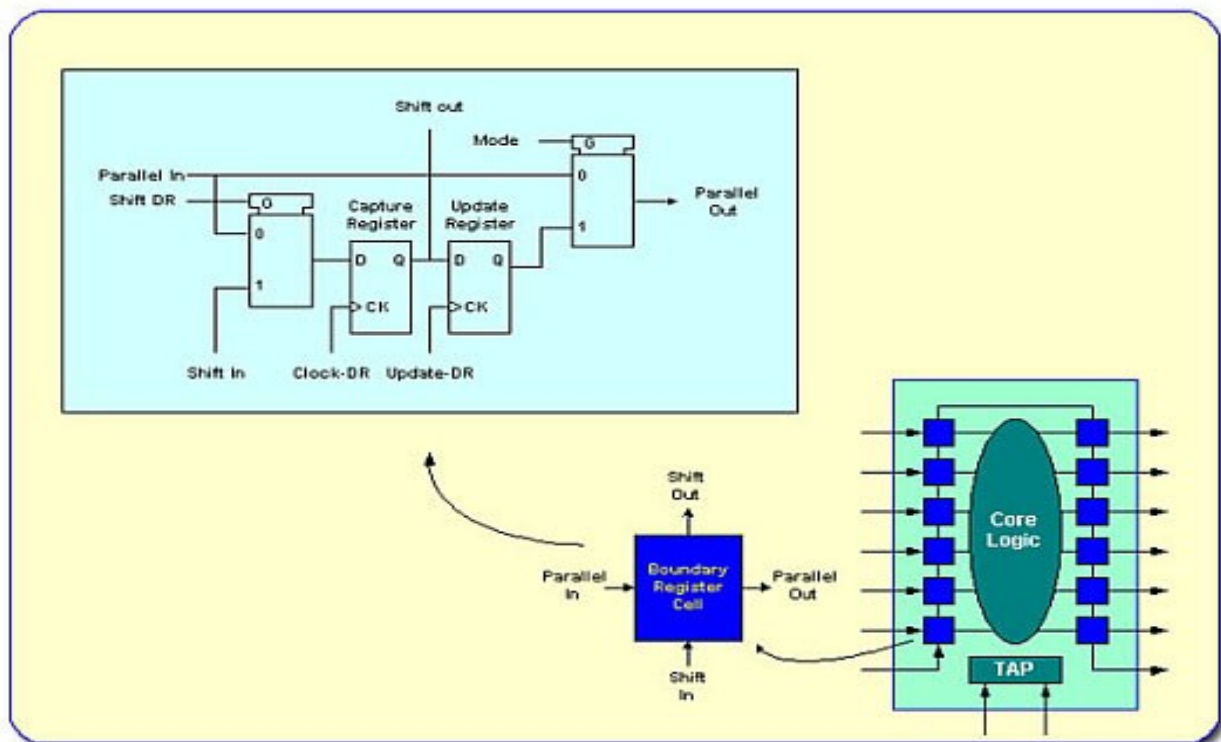
- Testovací logika se skládá s boundary-scan registru a dalších vestavěných bloků. Propřístup k testovací logice a jejímu řízení slouží tzv. **Test Access Port (TAP).**
- Obvody vyhovující normě JTAGu dovolují, aby do jejich komponent byly načteny testovací instrukce a data a po provedení testovacích instrukcí pak provést výstup výsledků testování.
- Komunikace s JTAGem (přenos instrukcí, testovacích dat a výsledku testů) probíhá sériově.
- Testovací logika je navržena tak, že sériový posuv instrukce neovlivní ty bloky obvodu, jejichž činnost je ovládána touto instrukcí. Vliv instrukce na tyto bloky se projeví až po dokončení posuvu instrukce (po jejím celém načtení do obvodu).
- Jakmile je jednou instrukce načtena, vybraný testovací obvod je nakonfigurován k odezvě. Nicméně v některých případech je potřeba do vybraného testovacího obvodu načíst příslušná data, aby výsledky testu byly smysluplné.
- Data jsou do obvodu načítána stejným způsobem jako instrukce, jejich načítání však neovlivní instrukci, která již byla do obvodu načtena předtím.
- Po provedení instrukce jsou pak data sériově vysouvána ven z obvodu.



Co je to Boundary-Scan ?

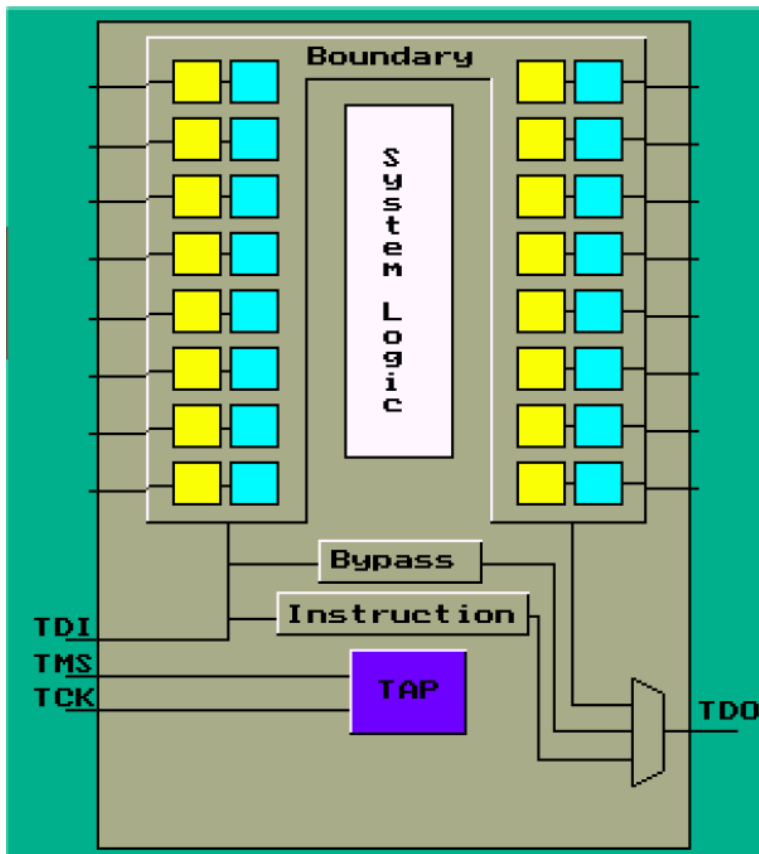
Základní myšlenka metody Boundary-Scan spočívá ve vložení jednoho článku posuvného registru mezi funkční bloky integrovaného obvodu a jeho vývody. Tento základní stavební prvek budeme nazývat Boundary-Scan "buňkou".

- Tento prvek je natolik univerzální, že může být použit na vstupním i výstupním vývodu integrovaného obvodu.
- Představme si např. že tato buňka je připojena na výstupní pin integrovaného obvodu.
- Signálový vstup pak představuje výstup funkční logiky obvodu a na signálový výstup je připojen na výstupní pin integrovaného obvodu.
- Boundary-scan buňky na pinech komponenty integrovaného obvodu jsou pomocí signálů sériový vstup sériový výstup propojeny do série a tvoří tak posuvný registr označovaný jako scanovací kanál.

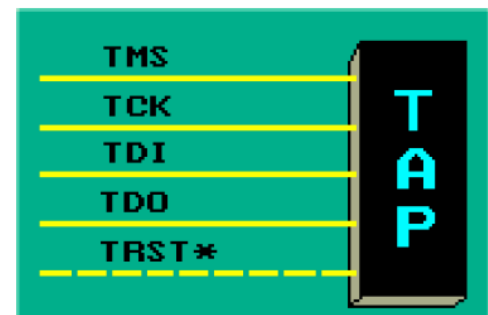


Test Access Port (TAP)

Pro přístup k testovacím funkcím vestavěným do obvodu slouží tzv. Test Access Port. Je sestaven minimálně ze tří vstupních a jednoho výstupního signálu.



Test Mode Select
Test Clock
Test Data In
Test Data Out
(nepovinný Test ReSeT)



- TCK - Test Clock Input - Vstupní signál TCK je do TAP zařazen proto, aby seriová datová cesta mezi komponentami zařízení mohla být používána nezávisle na hodinách jednotlivých komponent, protože ty se mohou v každé komponentě lišit. Tento způsob také dovoluje, aby posuv testovacích dat probíhal paralelně s činností komponent zařízení.
- TMS - Test Mode Select Input - Tento vstupní signál je dekodován řadičem TAP a slouží k výběru testovacích operací. Pomocí hodnot TMS jsou řízeny přechody mezi stavy ve stavovém diagramu řadiče TAP. Hodnota na vstupu TMS je snímána při náběžné hraně TCK.
- TDI - Test Data Input - Datový vstup TDI a datový výstup TDO se používají k seriovému pohybu testovacích dat obvodem. Hodnoty ze stupu TDI jsou načítány do příslušného datového nebo instrukčního registru, který se vybrán aktuální instrukcí. Hodnota na vstupu TDI je snímána při náběžné hraně TCK.

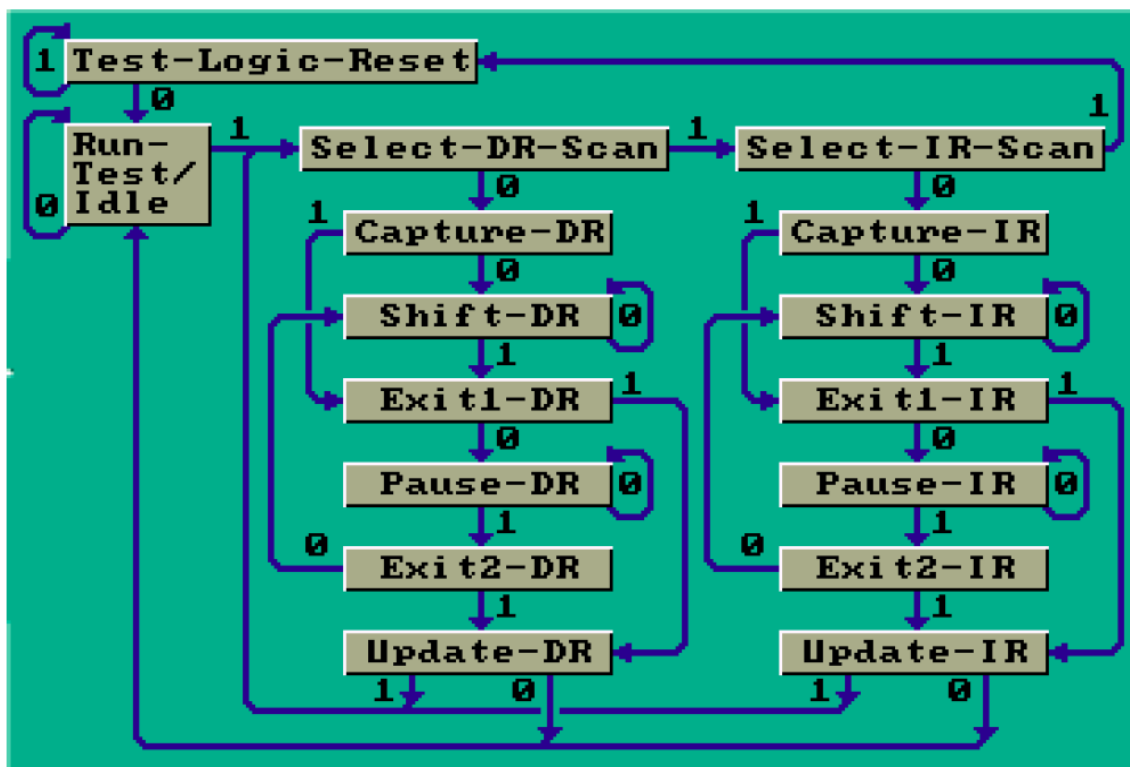
- TDO - Test Data Output - Hodnota na vstupu TDO je platná při sestupné hraně TCK.
- TRST - Test Reset Input - Tento signál je nepovinný. Pokud je přítomen, pak pokud je jeho hodnota rovna logické nule, musí řadič TAP přejít asynchronně do stavu Test-Logic-Reset. Tento signál není povinný proto, že inicializaci (resetování) řadiče TAP můžeme dosáhnout i jinak a to buď aplikací logické jedničky na signál TMS po dobu minimálně pěti náběžných hran TCK nebo zapnutím napájení integrovaného obvodu..

Je-li na desce více integrovaných obvodů vybavených JTAG rozhraním, je možno zřetěžit je sériově prostřednictvím řídicích signálů rozhraní (Obr. 3). Tímto uspořádáním může být prováděna nejen diagnostika jednotlivých obvodů, ale i jejich vzájemného propojení.

Řadič TAP

Řadič TAP synchronní konečný automat, který řídí činnost testovací logiky uvnitř integrovaného obvodu v závislosti na signálu TMS. Všechny přechody jsou uskutečňovány v závislosti na hodnotě signálu TMS při náběžné hraně hodin TCK.

Stavový diagram TAP kontroleru



Popis jednotlivých stavů:

- **Test-Logic-Reset** - V tomto stavu je testovací logika zablokována a nijak neovlivňuje normální funkci integrovaného obvodu. Do tohoto stavu se může řadič TAP dostat z jakéhokoliv jiného stavu pokud se na vstupu TMS objeví logická jednička po dobu minimálně pěti náběžných hran TCK. V tomto stavu je řadič nachází také po zapnutí napájení nebo pokud je na vstupu nTRST logická nula.

- Run Test/Idle - V tomto stavu jsou testovací obvody aktivní pouze pokud je v instrukčním registru určitá speciální instrukce. Například instrukce RUNBIST způsobí, že dojde ke spuštění speciálního vnitřního testu, vestavěného do obvodu jeho výrobcem. Instrukce které v tomto stavu nespouštějí žádnou funkci nesmí změnit hodnoty v datových registrech (Idle).
- Select-DR-Scan - Jedná se o dočasný stav řadiče, ve kterém si všechny datové registry vybrané současnou instrukcí zachovají svůj předchozí stav. Pokud se v tomto stavu při náběžné hraně TCK na TMS objeví logická nula, pak je odstartována scanovací sekvence vybraných datových registrů. Instrukce se v tomto stavu nemění.
- Select-IR-Scan - Jedná se o dočasný stav řadiče, ve kterém si všechny datové registry vybrané současnou instrukcí zachovají svůj předchozí stav. Pokud se v tomto stavu při náběžné hraně TCK na TMS objeví logická nula, pak je odstartována scanovací sekvence instrukčního registru. Instrukce se v tomto stavu nemění.
- Capture-DR - V tomto stavu mohou být do datových registrů vybraných aktuální instrukční paralelně nahrána data . Jestliže vybrané registry nemají paralelní vstup, pak se jejich hodnota nezmění. Aktuální instrukce se v tomto stavu nemění.
- Shift-DR - V tomto stavu všechny registry, které byly spojeny (jako výsledek aktuální instrukce do datové cesty mezi TDI a TDO provedou, při každé náběžné hraně TCK, posuv o jeden bit směrem k TDO. Ostatní datové registry, které nebyly zapojeny do datové cesty zůstanou nezměněny. Aktuální instrukce se v tomto stavu také nezmění.
- Exit1-DR - Jedná se o dočasný stav řadiče. Pokud se v tomto stavu při náběžné hraně TCK na TMS objeví logická jednička, pak je ukončena scanovací sekvence datových registrů. Datové registry ani aktuální instrukce se v tomto stavu nemění.
- Pause-DR - Tento stav řadiče dovoluje, aby posuv dat mezi TDI a TDO byl dočasně pozastaven. Datové registry ani aktuální instrukce se v tomto stavu nemění.
- Exit2-DR - Jedná se o dočasný stav řadiče. Pokud se v tomto stavu při náběžné hraně TCK na TMS objeví logická jednička, pak je ukončena scanovací sekvence datových registrů. Datové registry ani aktuální instrukce se v tomto stavu nemění.
- Update-DR - Některé datové registry mohou být vybaveny blokováním paralelním výstupem, aby se zabránilo změně paralelního výstupu v době posouvání dat. Paralelní výstupy datových registrů mohou být uvolněny pouze v tomto stavu nebo při vlastním testu zabudovaném výrobcem. Datové registry ani aktuální instrukce se v tomto stavu nemění.

- Capture-IR - V tomto stavu je do posuvného registru obsahujícího instrukci, nahrán vzorek logických hodnot. Přitom do nejnižšího bitu (bit nejbližší k sériovému výstupu) musí být uložena hodnota 1 a do druhého nejnižšího bitu hodnota 0. Toto opatření je zde z toho důvodu, aby se dala odhalit chyba ve scanovacím kanále. Do ostatních bitů mohou být nahrány hodnoty podle potřeby návrháře obvodu. Datové registry ani aktuální instrukce se v tomto stavu nemění.
- Shift-IR - V tomto stavu je posuvný registr obsahující instrukci připojen mezi TDI a TDO a posouván o jeden bit směrem k TDO při každé náběžné hraně TCK.
- Exit1-IR - Jedná se o dočasný stav řadiče. Pokud se v tomto stavu při náběžné hraně TCK na TMS objeví logická jednička, pak je ukončena scanovací sekvence instrukčního registru. Datové registry ani aktuální instrukce se v tomto stavu nemění.
- Pause-IR - Tento stav řadiče dovoluje, aby posuv instrukce mezi TDI a TDO byl dočasně pozastaven. Datové registry ani aktuální instrukce se v tomto stavu nemění.
- Exit2-IR - Jedná se o dočasný stav řadiče. Pokud se v tomto stavu při náběžné hraně TCK na TMS objeví logická jednička, pak je ukončena scanovací sekvence instrukčního registru. Datové registry ani aktuální instrukce se v tomto stavu nemění.
- Update-IR - Pouze v tomto stavu může být uvolněn paralelní výstup instrukčního registru a nová instrukce se stává aktuální instrukcí.

Instrukce

Instrukční registr dovoluje, aby do něj byla sériově nahrána instrukce, která pak řídí funkci testovací logiky. V této kapitole se seznámíme se základními instrukcemi, které musí mít každý obvod vyhovující specifikaci JTAG.

- Každá instrukce musí přesně určovat skupinu datových registrů, které se budou při provádění této instrukce používat.
- Ostatní datové registry musí být řízeny tak, aby neovlivňovali činnost systémové logiky a datové registry, které se používají.
- Každá instrukce musí datové registry, které se používají propojit do datové cesty mezi TDI a TDO.
- Kódy instrukcí, které nejsou nějakým způsobem využívány musí být ekvivalentní instrukci BYPASS.

Instrukce

Povinné

- Extest
- Sample/Preload
- Bypass

Volitelné

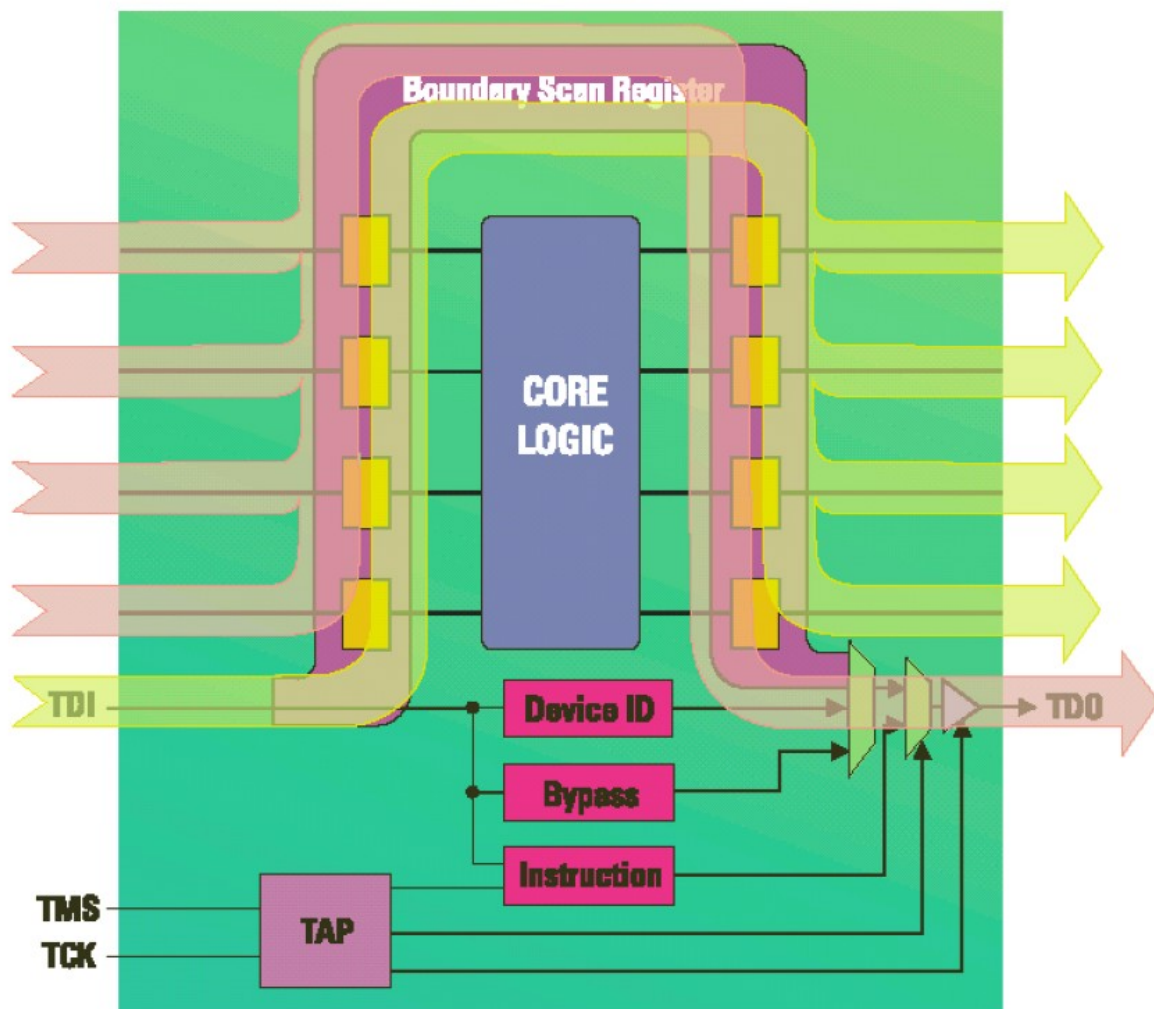
- IDcode
- Intest
- RunBIST
- Clamp
- HighZ

Tři základní instrukce, které musí mít každý obvod vyhovující JTAG:

- Instrukce BYPASS - Instrukční kód této instrukce musí být {111.1}, tedy ve všech buňkách instrukčního registru musí být logická jednička. Provedení instrukce BYPASS tedy můžeme dosáhnout tak, že na vstup TDI budeme přivádět logickou jedničku po celou dobu scanovacího cyklu instrukce. Tato instrukce používá při svém provádění pouze registr bypass, který připojí do datové cesty mezi TDI a TDO. Instrukce BYPASS se velmi dobře hodí, pokud máme na desce mnoho komponent a otestovat chceme např. jenom jednu. Ve všech komponentách, které nepotřebujeme testovat se jako aktuální instrukce nastaví instrukce BYPASS, která pro každou komponentu zařadí do datové cesty mezi TDI a TDO pouze jednobitový registr bypass a datová cesta pak bude mít délku počet komponent + počet vývodů komponenty, kterou chceme testovat. Bez použití této instrukce by byla délka datové cesty rovna součtu vývodu všech komponent, čímž by se doba testování rapidně a zbytečně prodloužila.

- Instrukce EXTEST - Instrukční kód této instrukce musí být {000.0}, tedy ve všech buňkách instrukčního registru musí být logická nula. Při použití instrukce EXTEST musí být stav všech signálů na výstupních pinech všech komponent definován daty nasunutými do boundary-scan registru (propojené boundary-scan buňky. Stav všech signálů na vstupních pinech všech komponent musí být uložen do boundary-scan registru ve stavu Capture-DR řadiče TAP. Tato instrukce používá při svém provádění pouze boundary-scan registr, který připojí do datové cesty mezi TDI a TDO. Instrukce EXTEST dovoluje, aby byla testována propojení mezi jednotlivými komponentami obvodu. Buňky boundary-scan registru na výstupních pinech komponent jsou použity jako budiče testovacích hodnot, zatímco buňky boundary-scan registru na vstupních pinech jsou použity k zaznamenání výsledků.

EXTEST



Nepovinné instrukce

- Instrukce IDCODE - Pokud komponenta obsahuje identifikační registr zařízení, pak musí mít tuto instrukci implementovanu. Při použití této instrukce je mezi TDI a TDO připojen pouze identifikační registr zařízení. Ve stavu Capture-DR je pak do tohoto registru paralelně nahrán identifikační údaj, který je pak možno vysunout na výstup TDO.
- Instrukce INTEST - Tato instrukce slouží k provedení testů systémové logiky a k získání její odezvy na tyto testy. Při této instrukci musí být stav všech vstupních a výstupních pinů komponenty určen daty nasunutými do boundary-scan registru. Ve stavu Capture-DR jsou pak do boundary-scan registru uloženy hodnoty na výstupu systémové logiky.
- Instrukce RUNBIST - Tato instrukce způsobí spuštění vnitřního testu komponenty vestavěného výrobcem. Samotný test se provádí ve stavu Run-Test/Idle řadiče TAP. Výhodou použití této instrukce je, že můžeme paralelně ve všech komponentách obvodu spustit vestavěný test a otestovat tak všechny komponenty, aniž bychom do obvodu museli pouštět nějaká testovací data.
- Instrukce USERCODE - Tato instrukce je určena pro programovatelné komponenty typu FPGA a EEPROM. Instrukce dovolí vnějšímu testeru (PC) určit uživatelské programování logického obvodu / paměti
- Instrukce CLAMP - Tato přepojí vstup TDI na výstup TDO přímo jako bypass místo jednobitového bypassu, dále pak propojí všechny logické výstupy logického jádra na snímací registry Boundary scan buněk.
- Instrukce HIGZ - Tato instrukce nastaví všechny výstupy logického obvodu do stavu vysoké impedance.

YAGARTO Tools

<http://www.yagarto.de/>

Balíček open source nástrojů pro překlad a ladění programů pod OS Windows pro procesory řady ARM, zaměřený na Assembler a jazyk C/C++ (vychází z GCC)

- Nepotřebuje k funkci CYGWIN (používá přímo WIN API)
- Kooperuje s prostředím **Eclipse**

Je složen ze tří komponent, některé (překladač) lze používat samostatně

- JTAG debugger interface podporující *J-Link GDB* server nebo *Open On-Chip Debugger* (OpenOCD, open source)
- Binutils, Newlib, GCC compiler, GDB debugger
- Eclipse Platform Runtime Binary and Eclipse CDT