



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

Digitális rendszerek tervezése

FPGA áramkörökkel

Konfiguráció és JTAG interfész

Fehér Béla

Szántó Péter, Lazányi János, Raikovich Tamás

BME MIT

FPGA laboratórium

FPGA konfiguráció

- **Az FPGA programozhatóság előny/hátrány**
 - Egyszer, azután fix működés
 - Ha módosítani akarunk
 - SRAM mindig, minden induláskor
- **A programozás technológiai háttere**
 - Fejlesztéskor egyedi, gyártó függő
 - Speciális berendezés
 - Nyitott, ipari szabványok szerinti
 - Általános célú berendezés
 - Egyedi fejlesztőeszköz
 - Beágyazott eszköz/környezet

FPGA konfiguráció

- **SRAM FPGA-k konfigurációja**
 - Nincs indulási alapl működés
 - a felhasználó szempontjából
 - Tetszőleges számú újrakonfiguráció
 - Minimum, ahányszor bekapcsoljuk
- **Konfigurációs program mérete**
 - Spartan 3E család 581kbit – 6 Mbit
 - Virtex5 család 6Mbit – 82Mbit
- **Konfiguráció ideje nem elhanyagolható!**

FPGA konfiguráció

- **Felhasználói támogatás minimuma**
 - Legyen minél egyszerűbb, észrevétlenebb
 - Legyen gyors
 - Ne igényeljen extra erőforrásokat, költséget
 - Illeszkedjen a rendszerbe
- **Felhasználói támogatás magasabb szintje**
 - Minden reális felhasználói igényhez illeszkedjen
 - Speciális üzemmódok támogatása, mint működés alatti részleges konfiguráció módosítás

FPGA konfigurációs módok

- **A lehetséges üzemmódok**
- **A vezérlés szerint**
 - Autonóm, MASTER
 - Vezérelt, SLAVE
- **Adatformátum szerint**
 - Soros 1 bit
 - Párhuzamos 8 bit (/16/32 bit)
- **Interfész szerint**
 - Egyszerű, egyedi
 - Szabványos
 - Speciális

FPGA konfigurációs módok

- **Spartan 3E család**

- Master Serial
- Slave Serial
- Slave Parallel x8
- Master SPI
- Master BPI U/D x8
- JTAG

- **Virtex5 család**

- Master Serial
- Slave Serial
- Master Parallel x8 x16
- Slave Parallel x8, x16, x32
- Master SPI
- Master BPI U/D x8, x16,
- JTAG

Konfiguráció I/O igénye

- **JTAG konfigurációs és teszt interfész** 4
 - TMS, TCK, TDI, TDO
- **Dedikált fix konfigurációs lábak** 2
 - PROG_B, DONE
- **Többfunkciós lábak** 7 - 35
 - MODE[2:0], CCLK, INIT_B, DIN,DOUT,
 - D[7:1], CSO_B, RDWR_B, A[23:0], LDC,HDC
- **A választott módtól függő**
 - Nem mindig veszteség, adott tervtől függ

BPI konfigurációs mód

- Egy teljes párhuzamos memória interfész, ~40 láb
- Eleinte volt, később kihagyták, ma ismét van
- Jellemző használati mód:
 - Korai FPGA-k kiegészítő logikai szerepet játszottak
 - Címdekóder, buszinterfész, illesztőegység, stb
 - Természetesen kapcsolódtak a rendszerbuszhoz,
 - Ugyanitt elérhetőek voltak a szokásos EPROM elemek
 - Az FPGA konfiguráció a „maradék„ területen elfért
 - Később mikrovezérlők integrálódtak
 - Saját belső buszinterfész, nem kell külső támogatás
 - FPGA HW egységek csak az adatinterfészt használták
 - 30 címvonal nagy áldozat → ami nem kell, ne legyen benne

BPI konfigurációs mód

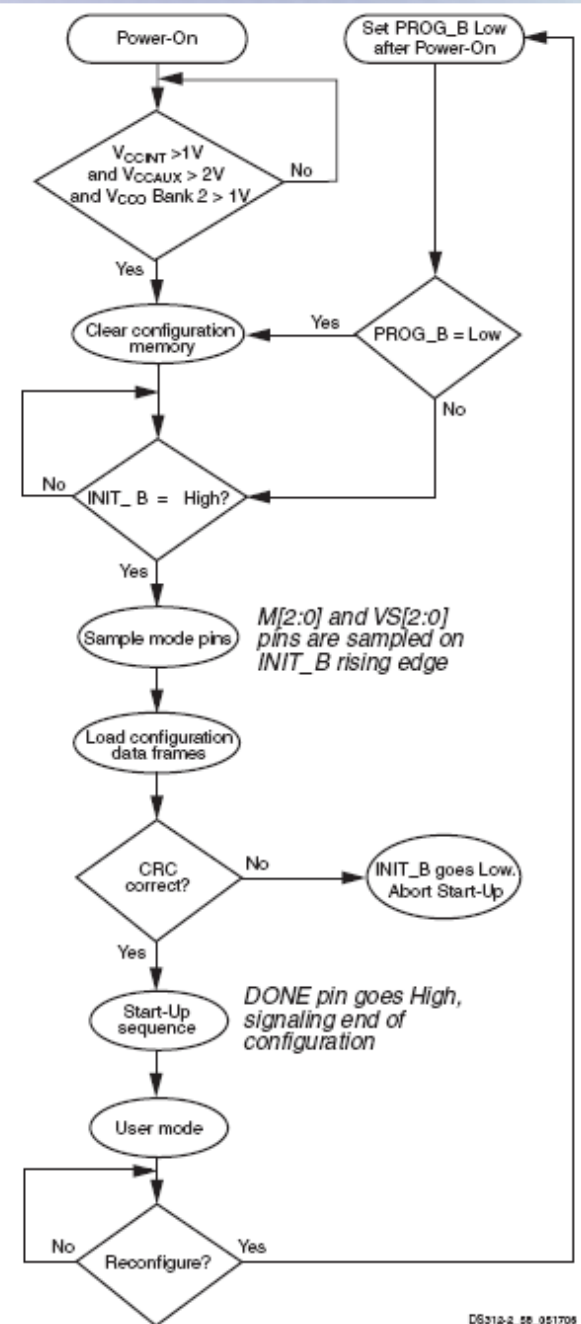
- **FPGA eszközök komplexitása nőtt**
- **Teljes mikroprocesszoros rendszer áramkörön belül → a program és adatmemória kivételével**
- **A memória kezeléshez adott számú I/O vonal kell**
 - De így már van értelme a konfigurációt egy normál párhuzamos flash memóriába tenni
 - Újra megjelent a Master BPI interfész, a nagyobb eszközöknél x16 opcióval együtt

FPGA soros konfiguráció

- **Az FPGA-ban a konfiguráció egy belső memóriába kerül**
 - Egyszerűbb esetben egyetlen hosszú shift regiszterbe léptetődik be az információ
 - Ez a legegyszerűbb adatátviteli mód
 - INIT_B után megfelelő számú CCLK + DIN
 - Kell egy külső soros PROM/Flash
 - FPGA eszközök sorba köthetők
 - 1 Master, a többi Slave

FPGA bekapcsolás

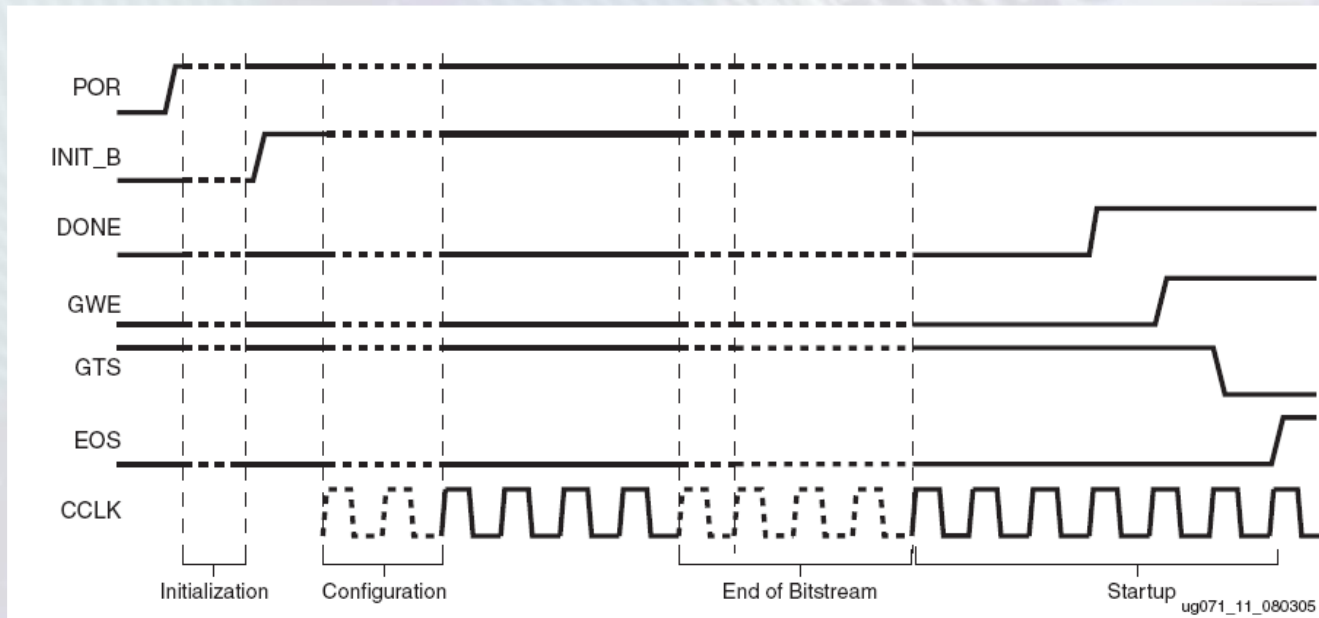
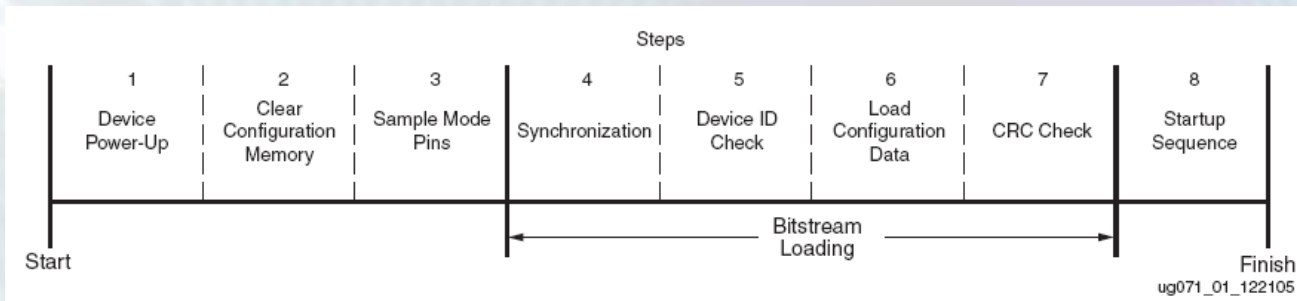
- A teljes folyamat lépései, kívülről érzékelve a lábak állapotát
- M[2:0] módjelek
- PROG_B vezérlőjel
- INIT_B, DONE státusz
- Megjegyzés:
 - A JTAG ettől függetlenül is használható



DS12-2_58_051709

FPGA bekapcsolás

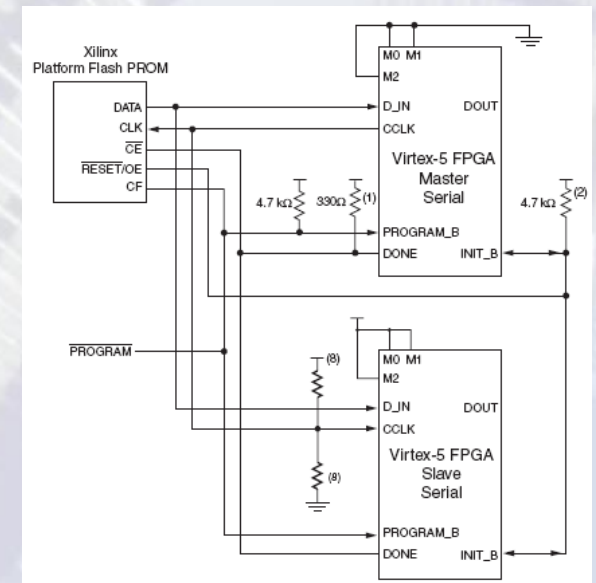
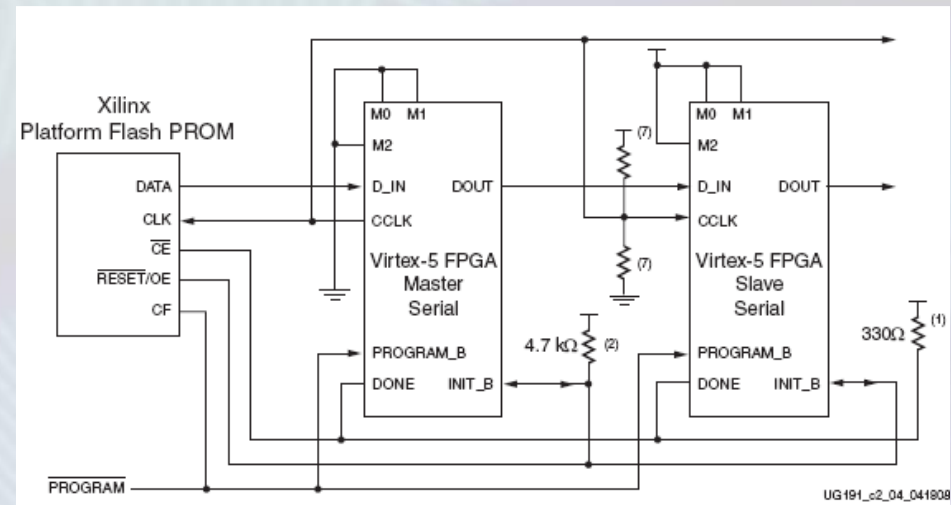
- Többfázisú folyamat, az előző diagram szerint



FPGA soros konfiguráció

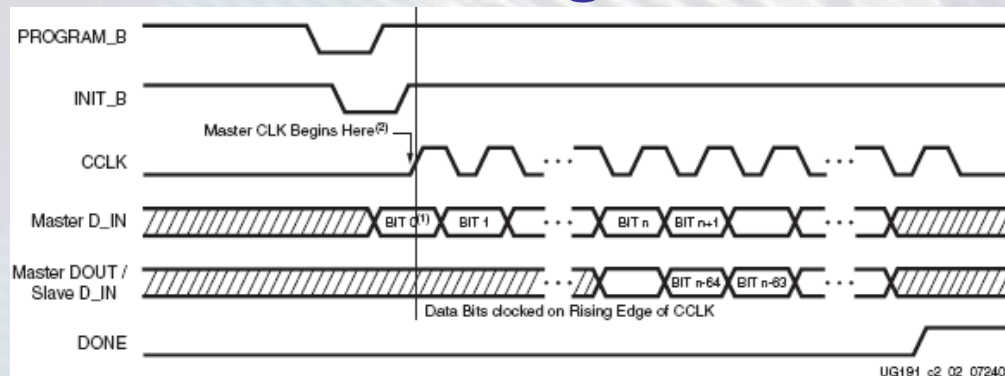
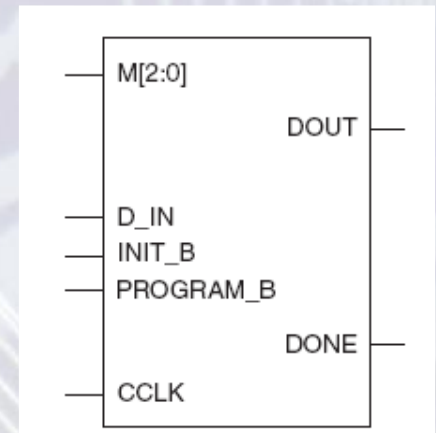
- Autonóm konfiguráció
- 1 Master, n Slave
- Speciális Xilinx PROM

- Láncba kapcsolva eltérő konfigurációk időben egymás után
- Párhuzamosan azonos konfiguráció minden eszközbe



FPGA soros konfiguráció

- Slave soros konfiguráció, pl. egy mikrovezérlőről
- 5 GPIO láb kell
- OUT PROG_B : Start Konfiguráció
- INP INIT_B: Teszt inicializálás
- OUT CCLK, DIN: Adatok beléptetése
- INP DONE: Teszt konfiguráció

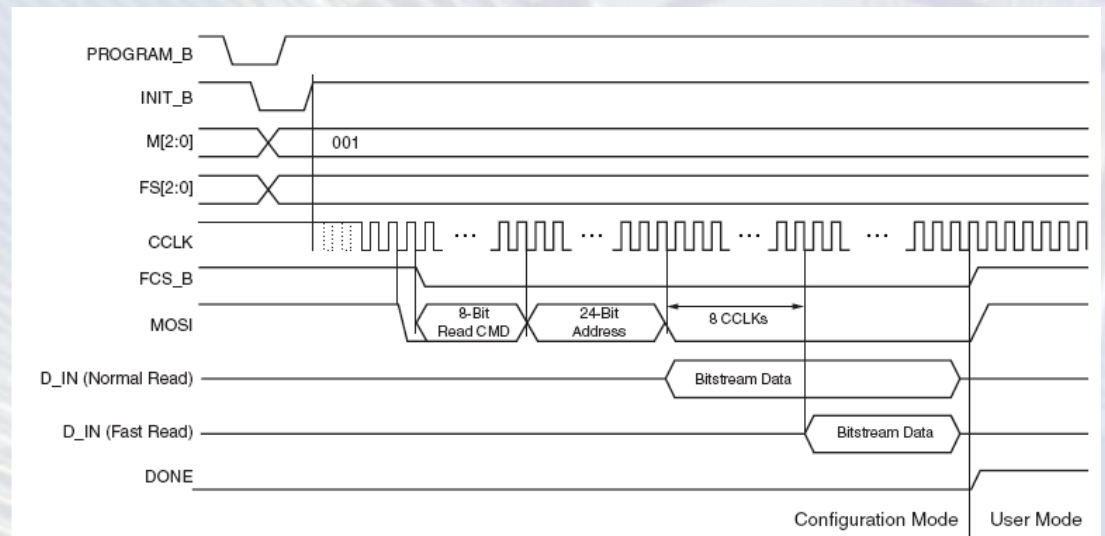
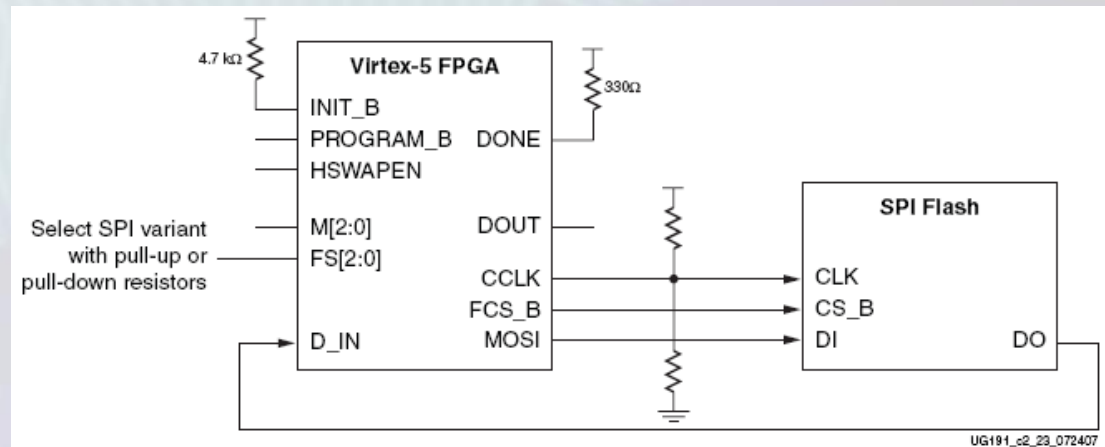


FPGA SPI konfiguráció

- **Hagyományos soros mód → speciális PROM**
 - Extra költség a felhasználónak,
 - Bevétel (+ extra gond) a gyártónak
- **Használjunk szabványos soros EPROM-ot, Flash-t**
- **SPI – Serial Peripheral Interface**
 - 4 vezeték, SCK, MOSI, MISO, CS_B
 - Ipari szabvány vezetékezés
 - Gyártófüggő parancskészlet és protokoll
 - Rugalmas, tetszőleges címezhetőség, de konfiguráció mindig 0 címtől!

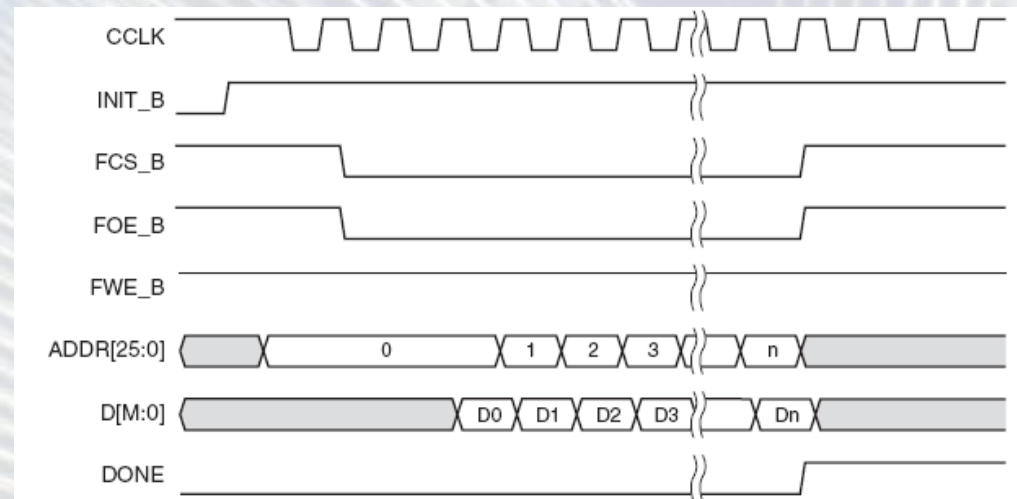
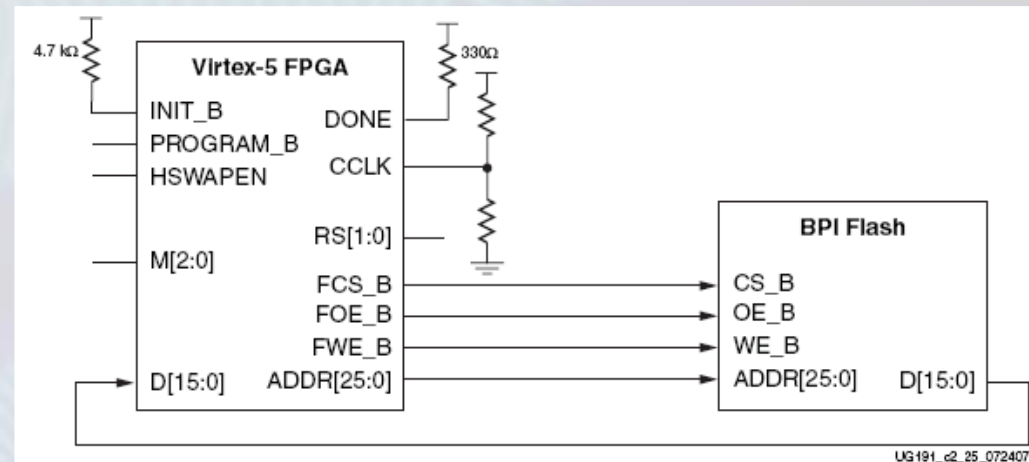
FPGA SPI konfiguráció

- Egyszerű interfész
- Nagyméretű SPI memóriák
- Maradék memória terület működés során használható (írás, olvasás is)
- Konfig terület az FPGA által újra írható → update!



FPGA BPI konfiguráció

- **Globálisan egyszerű, direkt kapcsolat egy párhuzamos flash memóriával (x8 x16)**
 - Címzés lehet UP vagy DOWN, a CPU RESET vektor értékétől függően
 - Adatbusz méret a fájl elején, 0xBB 0x11 vagy 0xBB 0x22

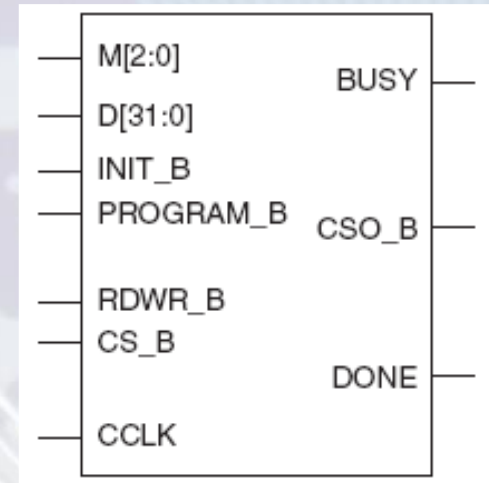


FPGA BPI konfiguráció

- **MultiBoot opció**
 - A külső memóriában több konfiguráció is lehet
 - Indulás a default adatokkal
 - Dedikált logika, konfiguráció újraindítása belülről az IPROG jellel
 - A konfiguráció kezdőcíme a WBSTART (Warm Boot Start Address Register)
 - IPROG generálható a belső konfiguráció vezérlőn keresztül

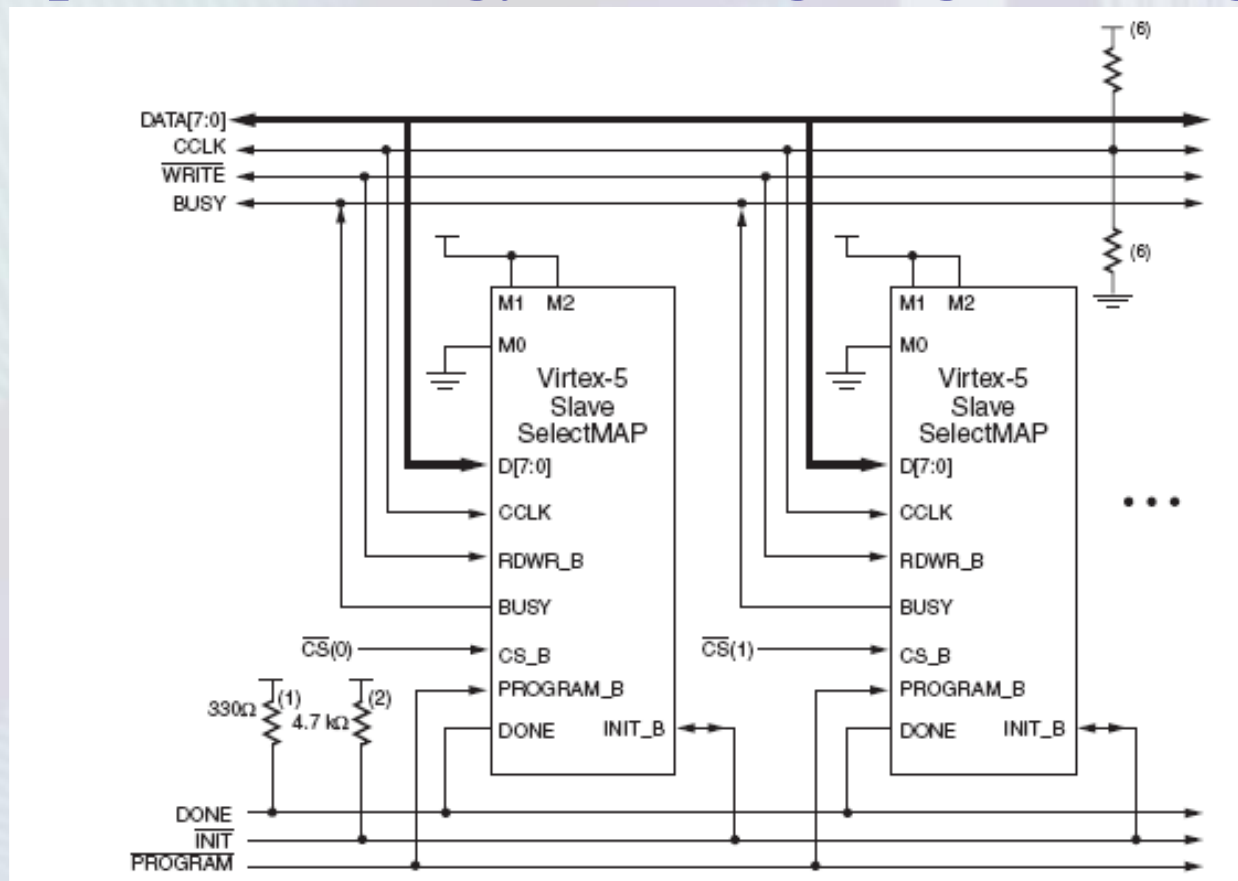
FPGA SelectMAP konfiguráció

- **Speciális bájtpárhuzamos interfész**
 - A normál soros x8, x16, x32 verziója
- **Konfigurálás után megtartható**
 - Konfiguráció visszaolvasása
 - Részleges dinamikus átkonfigurálás
 - Néhány speciális blokkhoz szükséges lehet
 - Órajel kezelő modul bizonyos beállításai
 - Nagysebességű soros interfész beállításai
 - Önmagában PAR-t használó alkalmazások



FPGA SelectMAP konfiguráció

- Gyakran használt beágyazott periféria interfész
- Egyszerű protokoll, nagy sebesség, rugalmasság
- CS_Bi

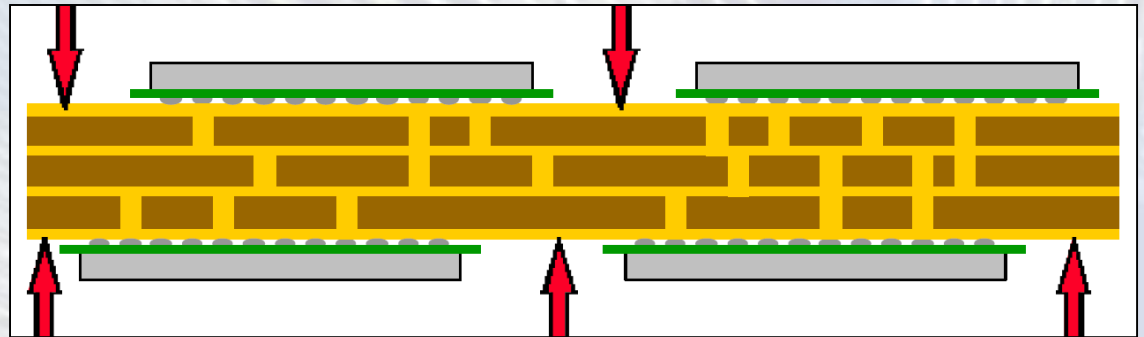
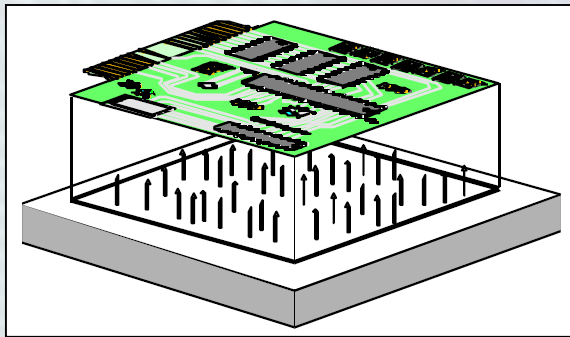


FPGA JTAG interfész

- **Kezdetben hiányzott, manapság a legelterjedtebb**
- **Előnye: virtuálisan nem igényel extra lábakat**
 - Elméletileg a JTAG interfész más okok miatt már része az eszköznek
 - Szabványos interfész, mindenki használja
 - Tesztelés, programozás, debug egyetlen egységes interfészen
 - Mindig aktív, mindig elérhető
 - Minden konfigurációval kapcsolatos műveletet támogat

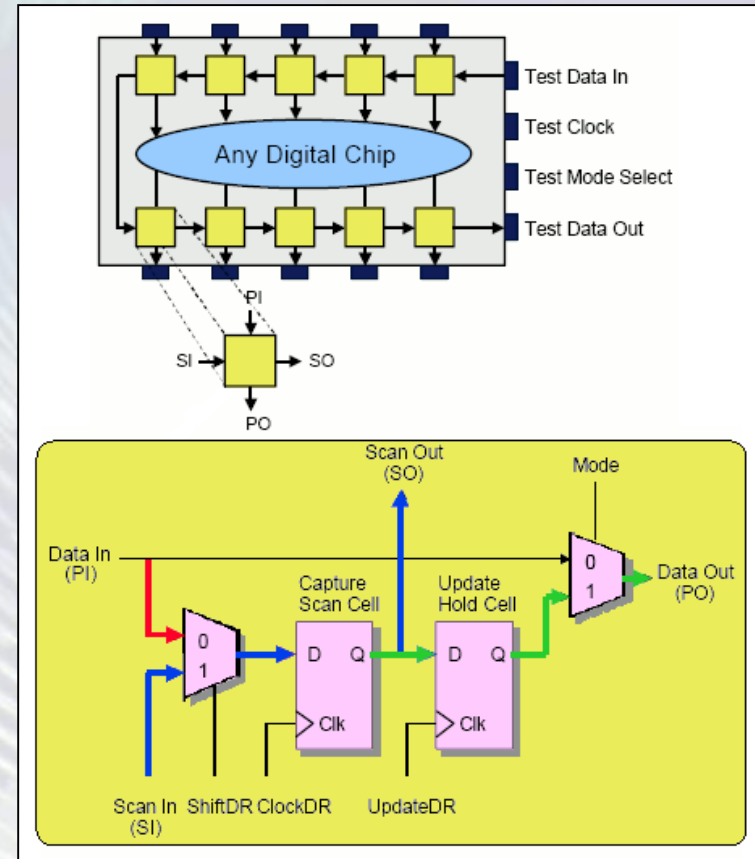
JTAG - Bevezetés

- **Digitális áramkörök tesztelése régen: tűágyas tesztelés**
 - Áramkörök közötti fizikai összeköttetések ellenőrzése
 - Működés ellenőrzése, funkcionális tesztelés
- **Méretetek csökkentése → új technológiák:**
 - Felületszerelt tokozású alkatrészek (QFP, BGA, stb.)
 - Többrétegű nyomtatott áramkörök (belső rétegek)
- **Következmény: a tűágyas tesztelés nem valósítható meg**
- **Megoldás: peremtesztelés (boundary scan)**



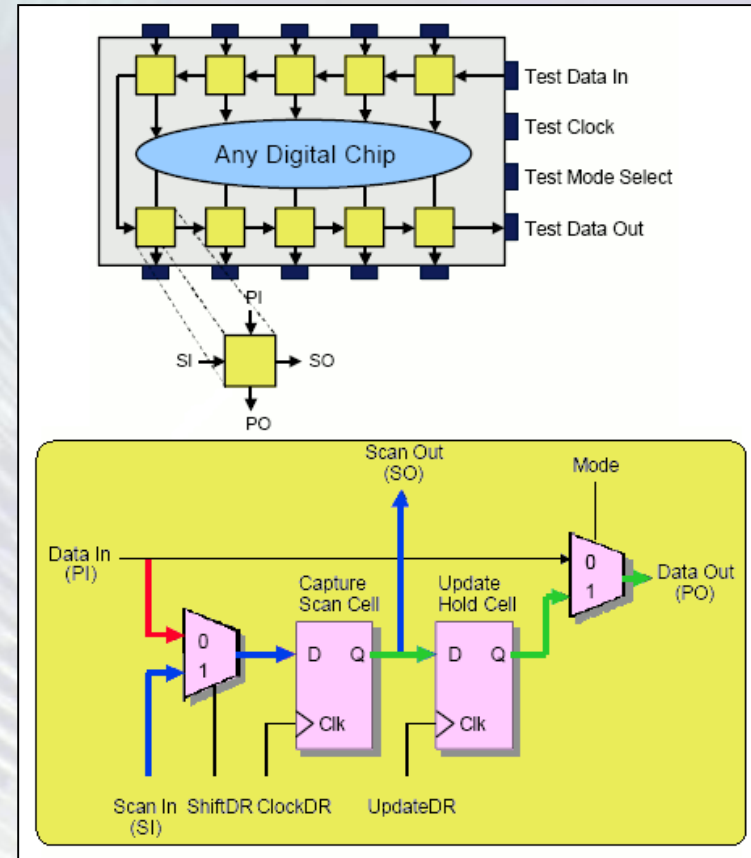
JTAG - Peremtesztelés

- **Regiszterek az I/O lábak és a chip között (boundary scan cellák):**
 - Bemenetek mintavételezése
 - Adatok kiírása a kimenetekre
- **A cellák felépítése:**
 - Többféle létezik
 - Az ábrán: BC-1 típusú cella
- **Soros adatvonalak: SI, SO**
 - Cellák közötti összeköttetés
 - Shiftregiszter megvalósítása
- **Párhuzamos adatvonalak: PI, PO**
 - A digitális áramkörhöz
 - A tokozás lábaihoz



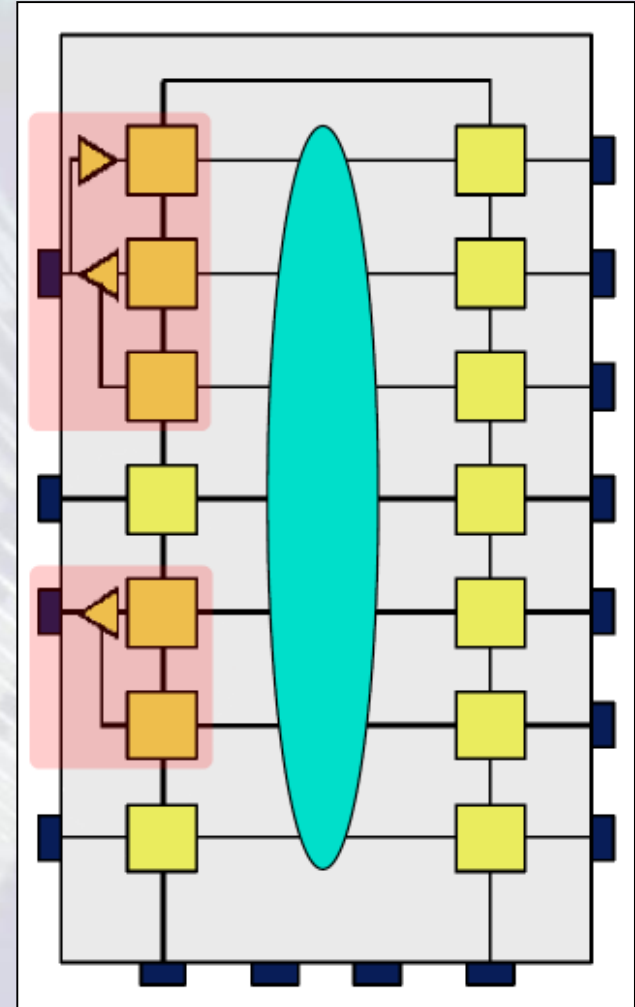
JTAG - Peremtesztelés

- **A boundary scan regiszter képes:**
 - A bemeneti jelek (PI) azonos időben történő mintavételezésére (CAPTURE)
 - A kimeneti jelek (PO) azonos időben történő frissítésére (UPDATE)
 - Soros adatok továbbítására a szomszédos cellának (SHIFT)
 - Transzparens működésre
- **A fenti műveletekhez a TAP vezérlő egy-egy állapota tartozik (lásd később)**



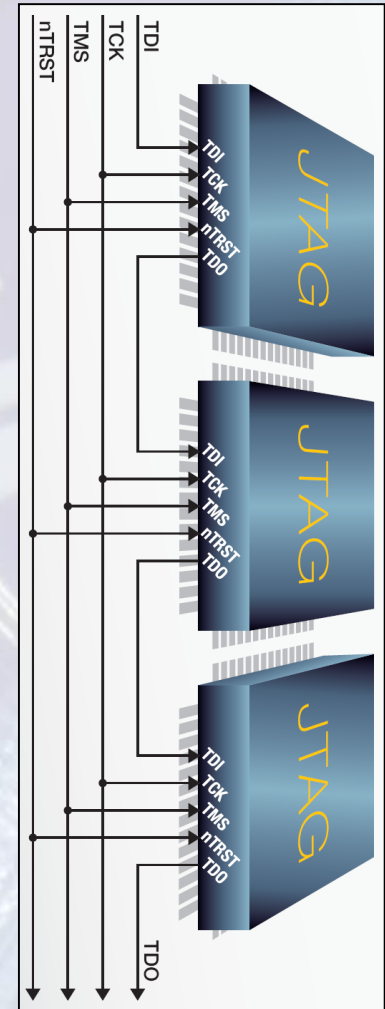
JTAG - Peremtesztelés

- **I/O lábak:**
 - Bemeneti adatvonal
 - Kimeneti adatvonal
 - Kimenet engedélyező jel
 - *3 szomszédos cellát igényel*
- **Nagyimpedanciás kimenetek:**
 - Kimeneti adatvonal
 - Kimenet engedélyező jel
 - *2 szomszédos cellát igényel*



JTAG – Fizikai interfész

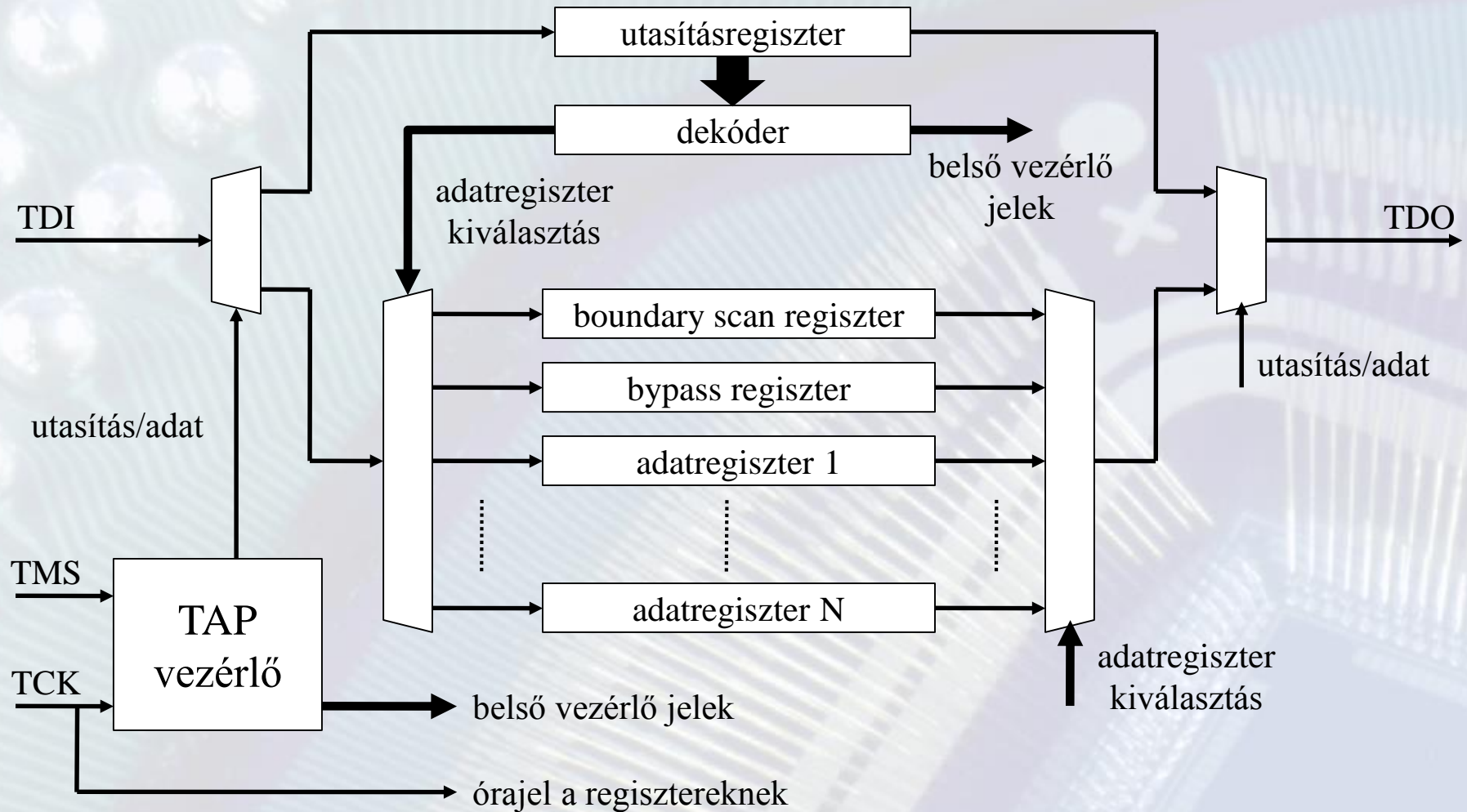
- **Egyszerű szinkron soros interfész:**
 - **TDI (Test Data Input):** soros bemenet
 - **TDO (Test Data Output):** soros kimenet
 - **TCK (Test Clock):** órajel
 - **TMS (Test Mode Select):** TAP kontroller vezérlésére szolgál
 - **TRST (Test Reset):** TAP kontroller reset (opcionális, sok eszközön nincs meg)
- **A soros interfészen keresztül az eszközök láncba fűzhetők (JTAG lánc):**
 - TDO → TDI
 - TCK, TMS és TRST közös



JTAG – Belső felépítés

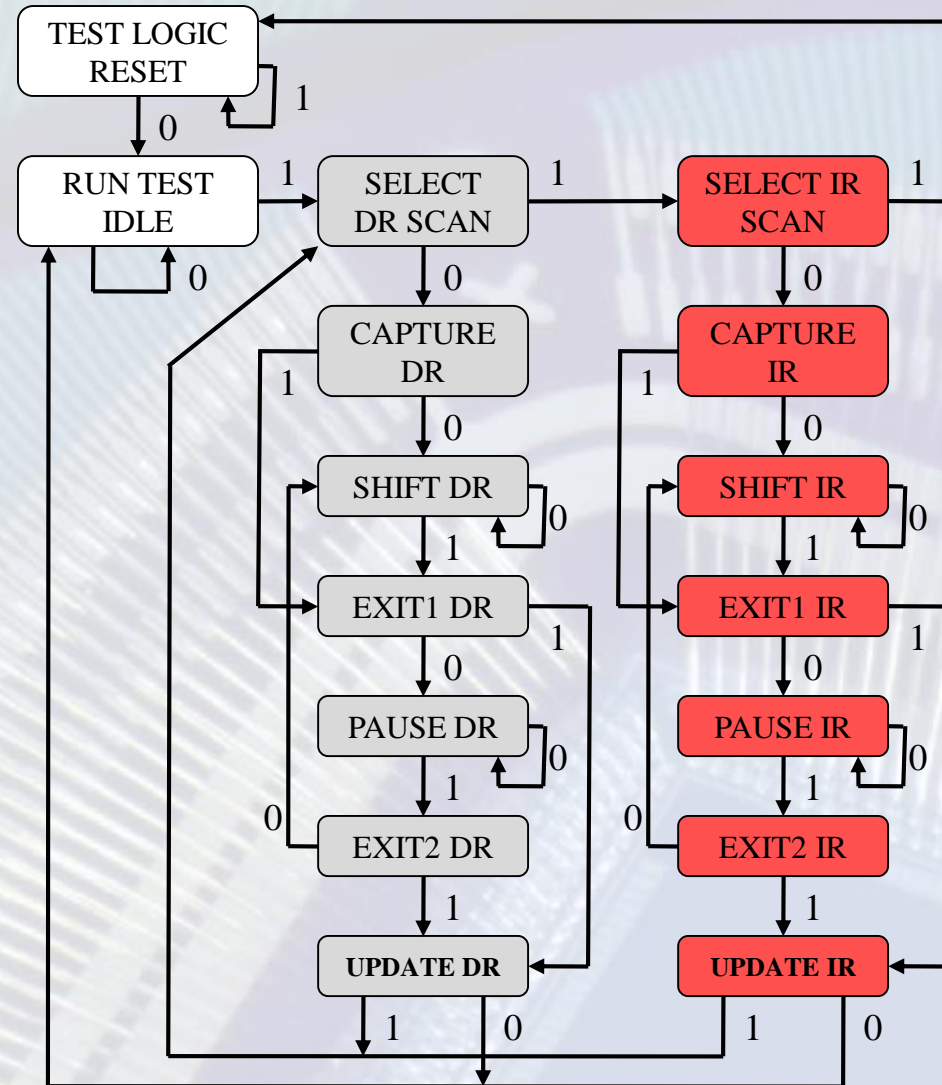
- **Minden JTAG interfészt implementáló IC tartalmaz:**
 - TAP (Test Access Port) vezérlőt
 - Belső regisztereket: utasításregiszter és adatregiszterek
- **Adatregiszterek:**
 - Boundary scan regiszter (kötelező)
 - Bypass regiszter (kötelező)
 - ID regiszter
 - Stb.
- **Programozható eszközök (mikrovezérlő, CPLD, FPGA): konfigurálás, debuggolás JTAG interfészen keresztül (is)**
 - Programozási, konfigurálási célú regiszterek
 - Debug regiszterek

JTAG – Belső felépítés



JTAG – TAP vezérlő

- 16 állapotú véges automata
- Az állapotátmenet a TMS jel vezérli (lásd a nyilakon)
- Regiszterek vezérlése:
 - Mintavételezés (CAPTURE)
 - Adatok shiftelése (SHIFT)
 - Adatok frissítése (UPDATE)
 - Transzparens működés
- Utasításregiszter elérése: piros állapotok
- Kiválasztott adatregiszter elérése: szürke állapotok

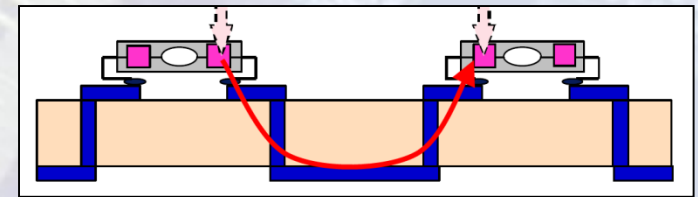
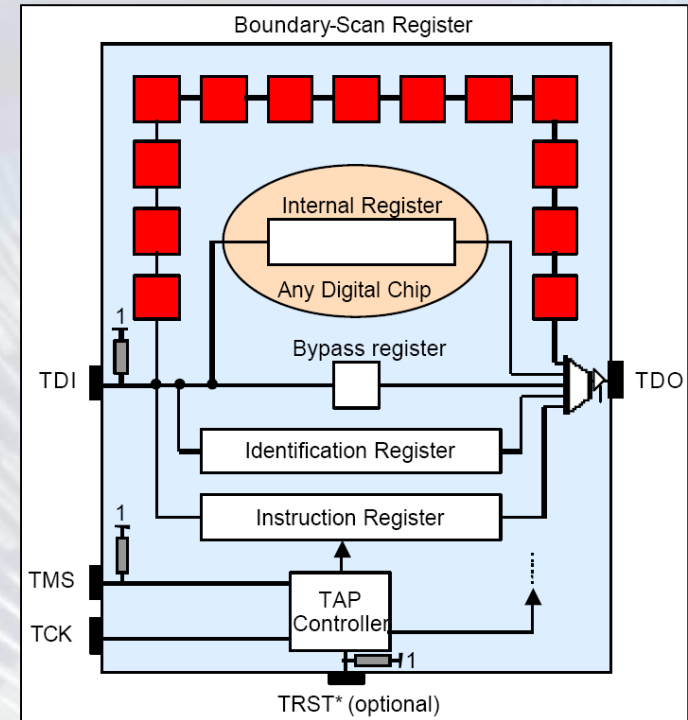
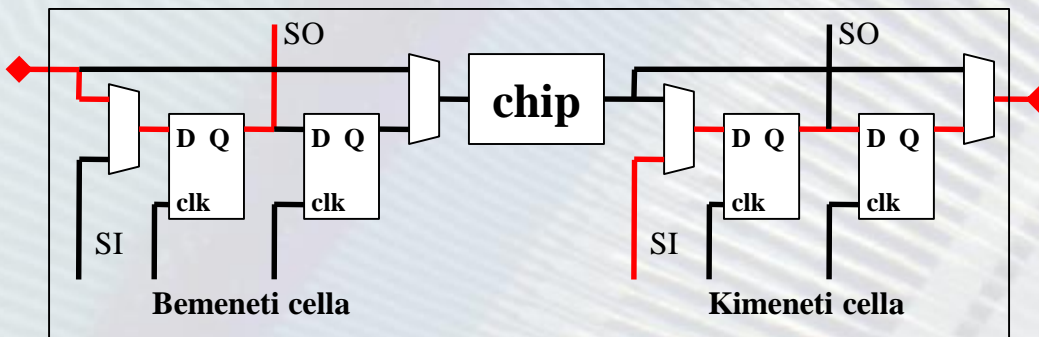


JTAG – TAP vezérlő

- **SHIFT-IR állapot: utasításregiszter feltöltése és kiolvasása**
- **3 kötelező utasítás:**
 - EXTEST, BYPASS, SAMPLE/PRELOAD
- **Egyéb utasítások:**
 - INTEST, IDCODE, USERCODE, RUNBIST
 - CLAMP, HIGHZ, USER1, USER2, stb.
 - Érvénytelen utasítások: BYPASS
- **Alapértelmezett utasítás a RESET állapotban:**
 - IDCODE, ha támogatott, egyébként BYPASS
- **Belső felhúzó ellenállások a TDI, TMS és TRST lábakon:**
 - Legfeljebb 5 TCK pulzus hatására RESET állapot
 - BYPASS utasítás kódja csupa 1-es

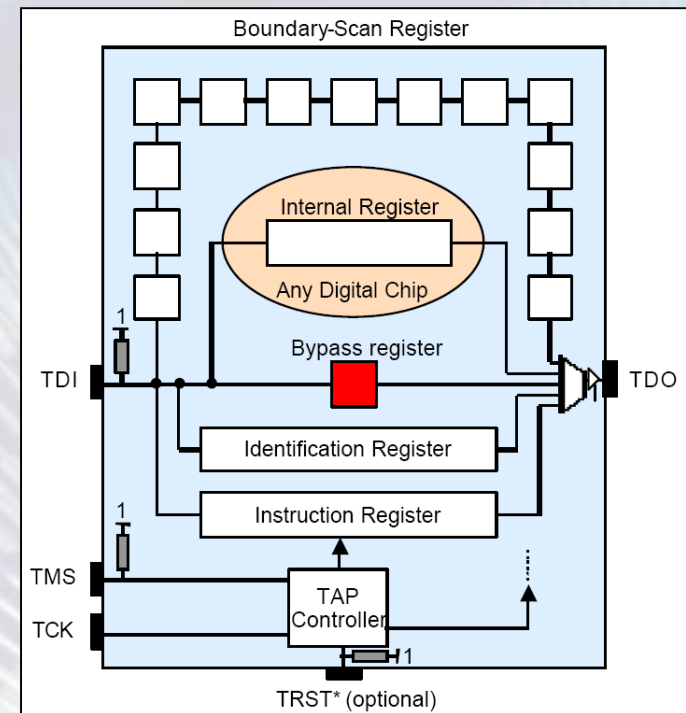
JTAG – EXTEST utasítás

- Kötelező utasítás
- Kiválasztott adatregiszter: boundary scan regiszter
- Áramkörök közötti összeköttetés tesztelése (EXternal TEST):
 - Adatok kiírása a kimeneti lábakra
 - A bemeneti lábak mintavételezése
- Az eszköz teszt módba kerül



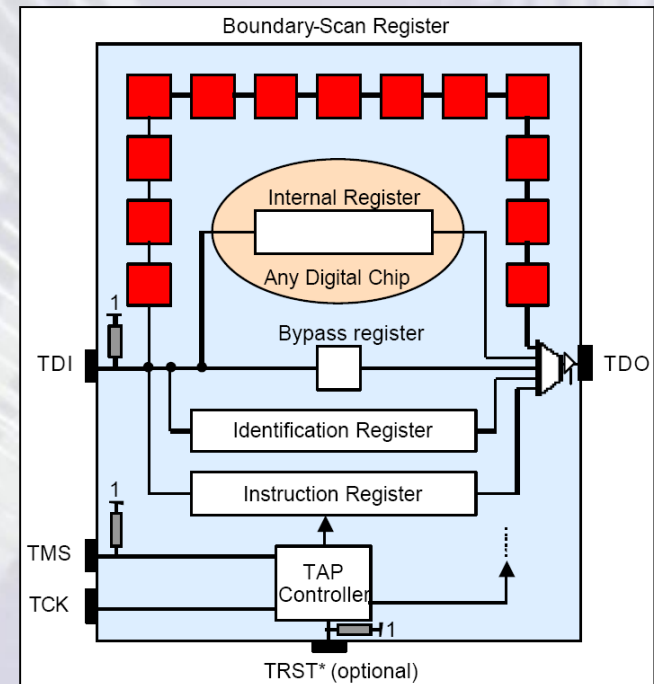
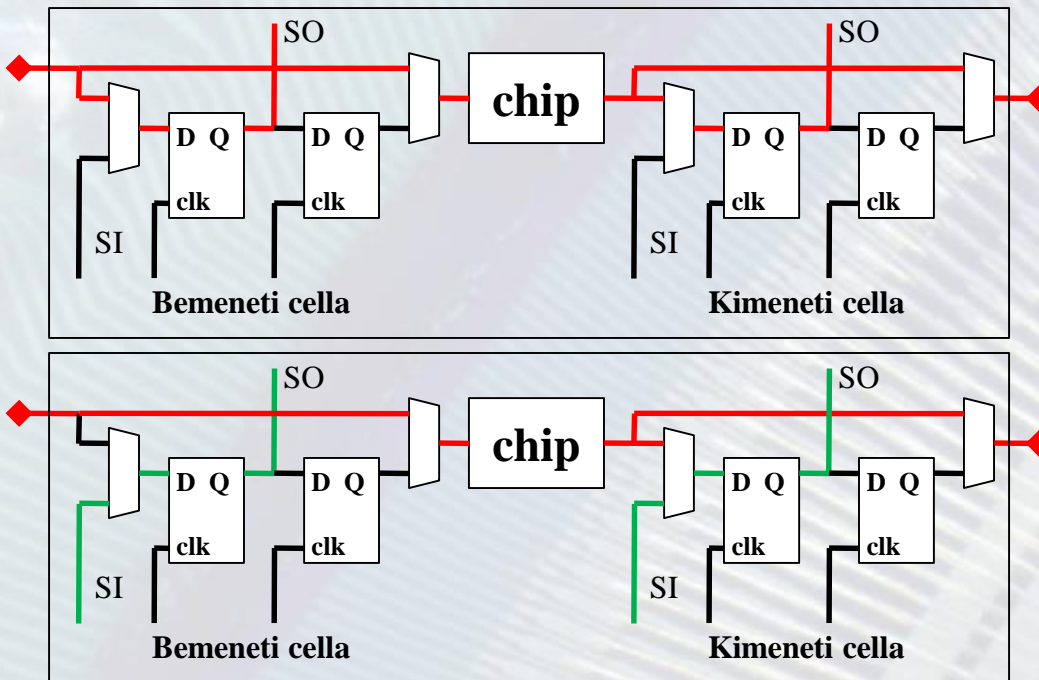
JTAG – BYPASS utasítás

- Kötelező utasítás
- A JTAG lánc lerövidítése
- A kiválasztott adatregiszter: bypass regiszter (1 bites)



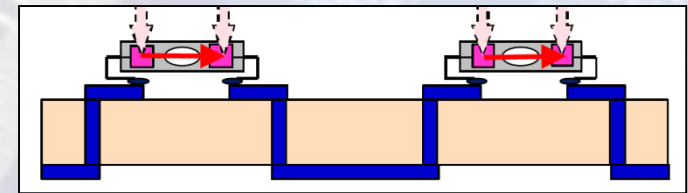
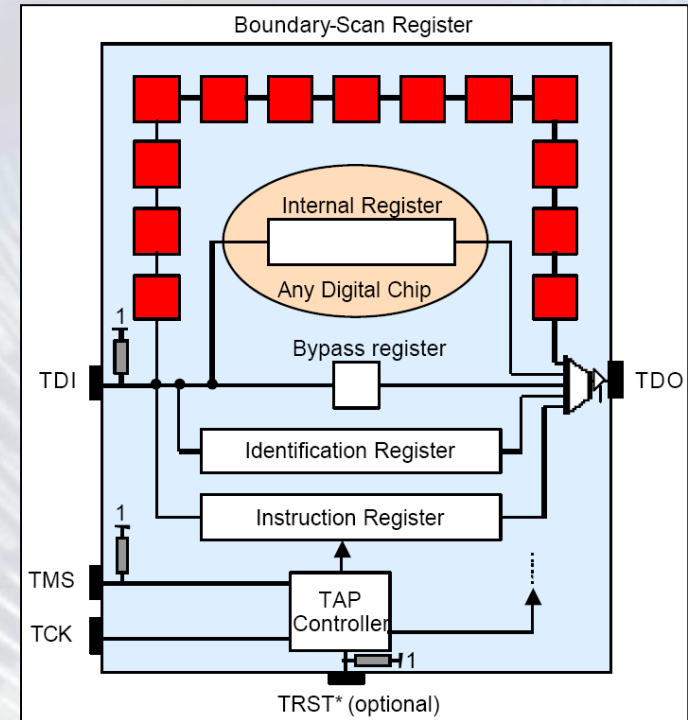
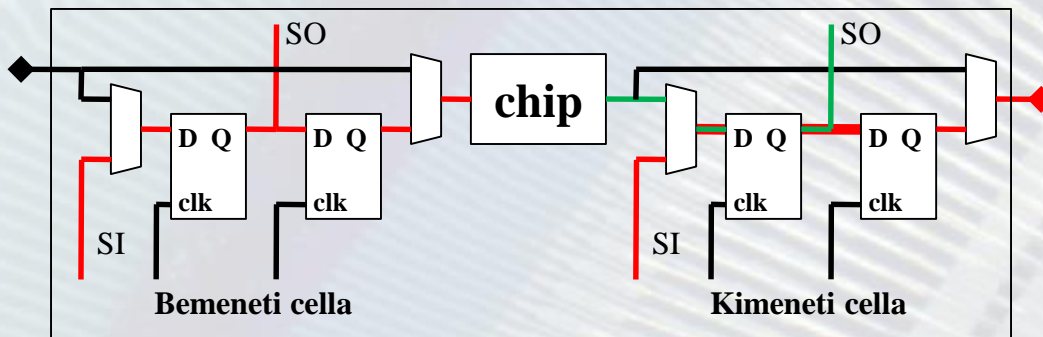
JTAG – SAMPLE/PRELOAD utasítás

- Kötelező utasítás
- Kiválasztott adatregiszter: boundary scan regiszter
- Jelek mintavételezése (bal felső ábra) és beolvasása (bal alsó ábra)
- Boundary scan regiszter feltöltése adott értékekkel (bal alsó ábra)
- Az eszköz normál működési módban marad



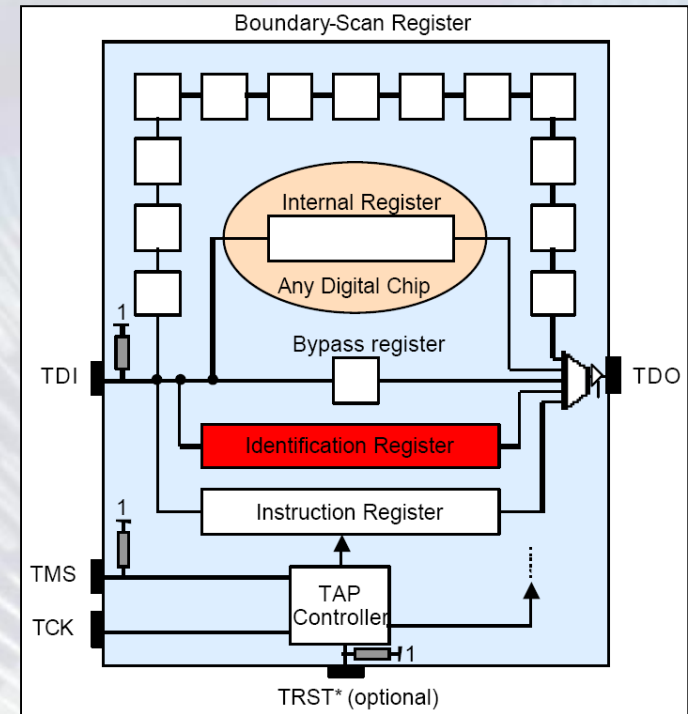
JTAG – INTEST utasítás

- Opcionális utasítás
- Kiválasztott adatregiszter: boundary scan regiszter
- Belső tesztelés (INternal TEST):
 - Tesztvektor beírása (piros)
 - Eredmények beolvasása (zöld)
- Az eszköz teszt módba kerül



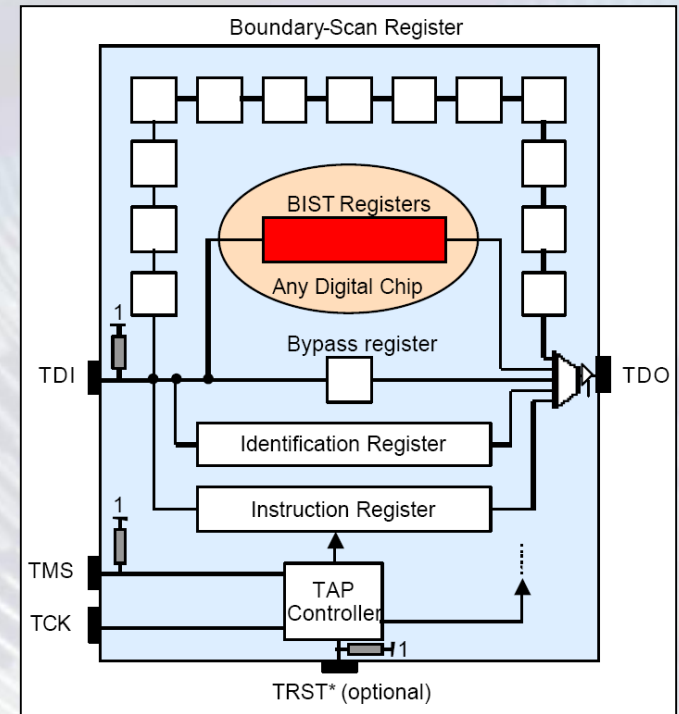
JTAG – IDCODE, USERCODE utasítás

- **Opcionális utasítások**
- **IDCODE utasítás:**
 - Kiválasztott regiszter:
32 bites ID regiszter
 - Az eszköz azonosító kód kiolvasása
- **USERCODE utasítás:**
 - Kiválasztott regiszter:
32 bites ID regiszter
 - Felhasználó által megadott azonosító kód kiolvasása
 - CPLD-k, FPGA-k esetén szokásos



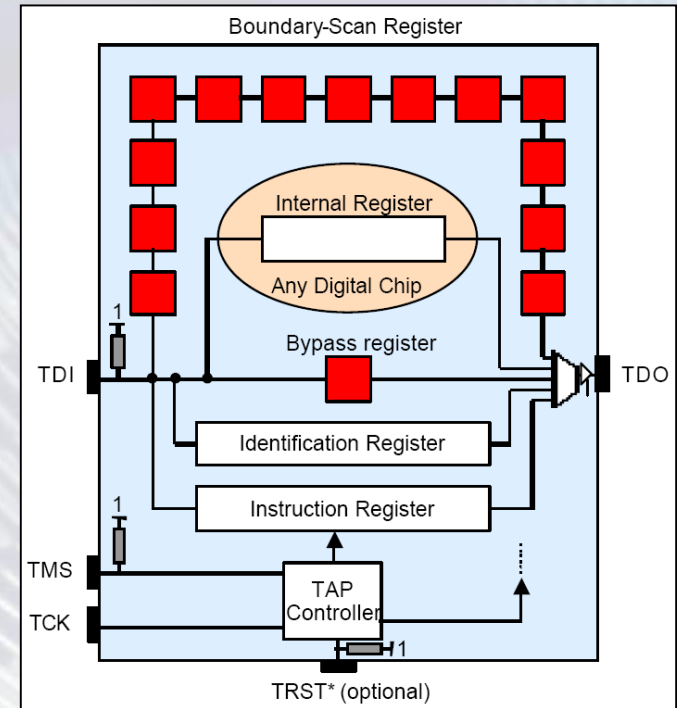
JTAG – RUNBIST utasítás

- Opcionális utasítás
- Kiválasztott adatregiszter: BIST regiszter
- Beépített önteszt elindítása
- Eredmények beolvasása



JTAG – CLAMP, HIGHZ utasítás

- **Opcionális utasítások**
- **CLAMP utasítás:**
 - A SAMPLE/PRELOAD utasítással korábban megadott értékek kerülnek a kimenetekre
 - BYPASS regisztert választja ki
- **HIGHZ utasítás:**
 - A kimenetek nagyimpedanciás állapotba kerülnek
 - BYPASS regisztert választja ki



JTAG – Egyéb utasítások

- **Programozható eszközök esetén:**
 - Programozási, konfigurálási utasítások
- **Processzorok, mikrokontrollerek esetén:**
 - Debug utasítások (sok esetben nem nyilvánosak)
- **FPGA-k esetén:**
 - Saját célú adatregiszterek is kialakíthatók
 - Ezek kiválasztására szolgálnak a USER utasítások

JTAG – SVF fájlok

- **SVF (Serial Vector Format):** szabványos szöveges fájlformátum
- A JTAG interfészen elvégzendő műveleteket írja le
- Egyszerű felépítésű, könnyen feldolgozható
- **Parancsok:**
 - *STATE*: TAP vezérlő állapotátmenet végrehajtása
 - *SIR*: utasításregiszter művelet végrehajtása
 - *SDR*: adatregiszter művelet végrehajtása
 - *ENDIR*: utasításregiszter művelet utáni végállapot megadása
 - *ENDDR*: adatregiszter művelet utáni végállapot megadása
 - Stb.
- **SVF specifikáció:**
<http://www.asset-intertech.com/support/svf.pdf>

JTAG – SVF fájlok

- A gyári fejlesztői környezetek lehetőséget biztosítanak a konfigurálást elvégző SVF fájl létrehozására
- Példa: az eszköz típusának ellenőrzése az ID kód alapján

```
// Created using Xilinx iMPACT Software [ISE Foundation - 7.1.04i]
TRST OFF;
ENDIR IDLE;
ENDDR IDLE;
STATE RESET IDLE;
FREQUENCY 1E6 HZ;
//Loading device with 'idcode' instruction.
SIR 6 TDI (09) SMASK (3f) ;
SDR 32 TDI (00000000) SMASK (ffffffff) TDO (f1414093) MASK (0ffffffff) ;
```

Ha a beolvasott ID kód nem 0x01414093, akkor jelezni kell a hibát és az SVF fájl további feldolgozását abba kell hagyni.

JTAG – BSDL fájlok

- **Boundary Scan Description Language**
- **VHDL nyelvre épülő formátum**
- **Adatokat tartalmaz az áramkörről, például:**
 - Kivezetések típusa (kimenet, bemenet, kétirányú)
 - Maximális TCK órajelfrekvencia
 - JTAG utasításregiszter hossza
 - A támogatott JTAG utasítások neve és kódja
 - Eszköz azonosító kód (ID regiszter tartalma)
 - Boundary scan regiszter felépítése:
 - A regisztert alkotó cellák típusa
 - A cellák és a tokozás kivezetéseinek összerendelése
- **A gyártók biztosítják az eszközökhöz a BSDL fájlokat**

FPGA JTAG konfiguráció

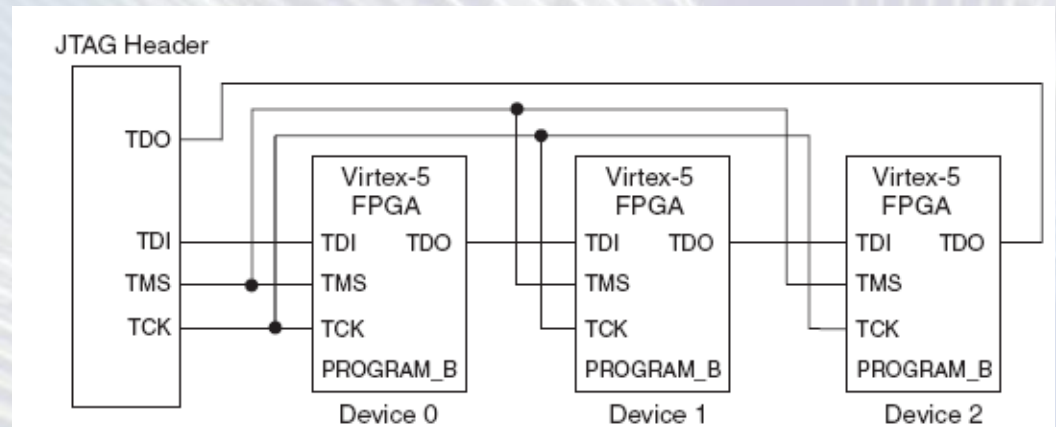
- Az FPGA eszközt a rendszerben lévő JTAG láncba illesztjük

– Ez elméletileg lehet tetszőlegesen komplex

- Eszközök száma
- Eszközök típusa

TCK jelvezetés NO_STAR !!!!
FPGA, flash, CPU, stb.

- Esetleg önálló JTAG kapcsolatot alakítunk ki



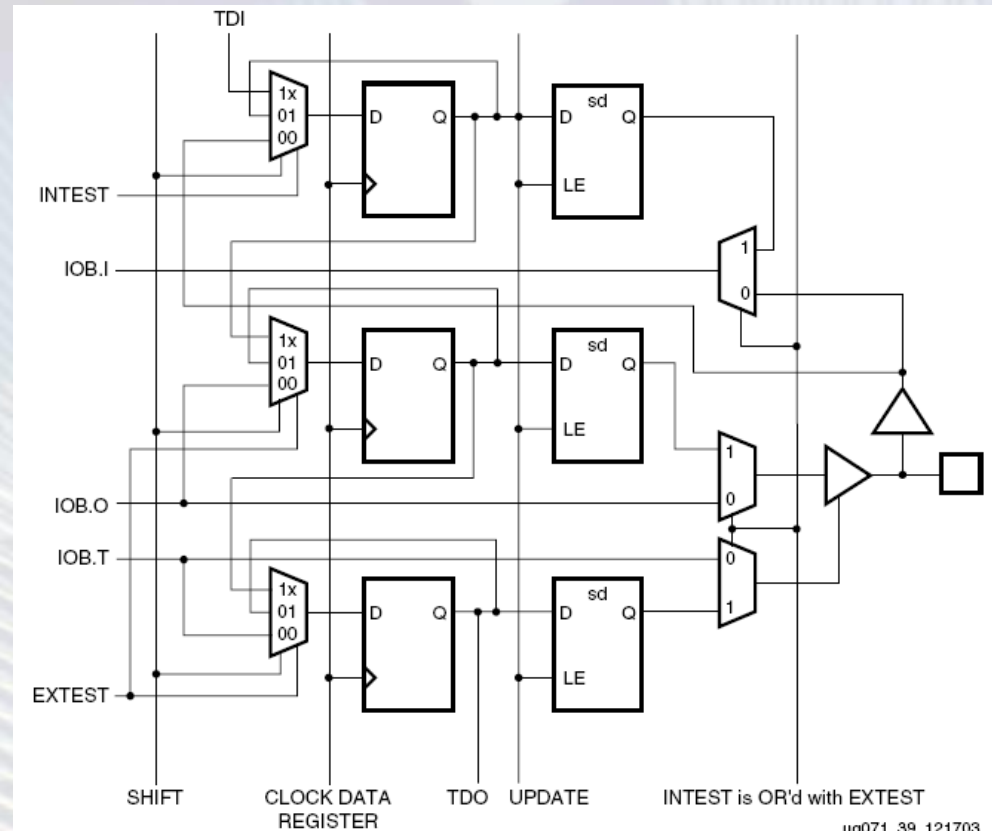
FPGA JTAG konfiguráció

- Gyártói fejlesztői környezetek ezt használják
- Egyszerű interfész, elfogadható sebesség
 - Eltérő feszültség szintek 1.8V – 3.3V
- IEEE1149.1 Alap JTAG szabvány
 - A programozás ezen belül lett kialakítva
- IEEE1532 A JTAG interfészre épülő programozás
 - Cél a hatékonyság növelése
 - Összetett protokoll, komplex parancsokkal

FPGA JTAG interfész

- **FPGA JTAG TAP vezérlő további opciói**
 - A kötelező regisztereken túl felhasználói regiszterek
 - USER1, USER2 (USER3, USER4)
 - Ezekkel saját interfész építhető a felkonfigurált FPGA felhasználói logikához
 - Tipikus példa a ChipScope Core, EDK MDB debug modul, de saját megoldás is kialakíthat

- **I/O cella IEEE 1149.1 Boundary Scan kiegészítése**
 - EZEK NEM AZ I/OB FF-ok, ez csak a BS logika



FPGA konfiguráció biztonság

- **SRAM eszközöknél felmerülhet a konfigurációs tartalom biztonsága**
- **Két eltérő szempont:**
 - Szellemei tulajdon védelme
 - Működés megbízhatósága
- **Mindkettő más megközelítést kíván**
 - Konfiguráció biztonságos kezelése, tárolása, betöltése
 - Konfiguráció gyakori ellenőrzése, visszaolvasása

FPGA konfiguráció biztonsága

- **Prémium eszközök (V4, V5) képesek titkosított konfigurációs bitfájl feldolgozására**
 - Dedikált 256 bites AES dekódoló logika
 - Csak a konfigurációs bitfájlhoz használható
 - A felhasználó nem éri el, nem látja
- **A folyamat**
 - A szoftver generálja a titkosított bitfájlt + a kulcsot
 - A kulcs JTAG-on keresztül feltölthető a speciális KEY SRAM memóriába, aminek független külső elemes táplálása van
 - Ezután a konfiguráció titkosított vagy nyitott bitfájlokkal is elvégezhető
 - Titkosított konfiguráció csak teljes lehet, utána nincs visszaolvasás, PAR, stb. Újrakonfigurálás csak teljes törlés, kikapcsolás után. (A kulcs megmarad!)

FPGA konfiguráció biztonsága

- **Az SRAM konfigurációs memóriában is előfordulhat a SEU jelensége (Single Event Upset)**
- **Védekezés ciklikus visszaolvasással**
 - A biztonság itt fordítva értelmezett
 - Akkor jó a rendszer, ha folyamatosan ellenőrizhető
 - Bármilyen probléma esetén újrakonfigurálás
- **Mi történik, ha a felhasználói logikában történik SEU egy FF-ban vagy RAM memória biten?**
 - Többségi logika, ellenőrző jelzések, stb.
 - Nagy megbízhatóságú rendszerek tervezési módszerei használhatók

Összefoglalás

- **Az FPGA konfiguráció a használat része**
 - Egyszer programozható eszközöknél nem rendszertехnikai probléma
 - SRAM elemeknél széles körű választási lehetőség a tervezési megoldások között
 - Elemi megoldás, csak induláskor kell konfigurálni
 - Összetettebb, szükséges új konfiguráció letölthetősége a működés során
 - Komplex, a működés része a konfiguráció váltás
 - Konfiguráció biztonsági kérdések