

JTAG

Lazar Premović 2019/0091

Šta je JTAG?

- Standard, originalno namenjen za testiranje sastavljenih štampanih ploča.
- Specifikuje fizički interfejs i serijski protokol za pristup TAP-u (*Test Access Port*).
- Standard kreiran 1985. od strane **Joint Test Action Group** (po kojoj je i dobio ime).
- 1990. postaje i IEEE standard (1149.1).

Motivacija i upotreba

- Vremenom uređaji (samim tim i njihove štampane ploče) postaju kompleksniji. Integrisana kola imaju sve veći broj pinova i oni postaju sve teže dostupni za testiranje.
- Potrebno je naći efikasniji način za testiranje ovakvih štampanih ploča.
- Rešenje: *Boundary Scan* (inicijalna motivacija za JTAG).
- Iako je JTAG inicijalno dizajniran i korišćen samo za boundary scan, kasnije je standard „proširen“ dodatnim mogućnostima kao što su ICD (*In Circuit Debugging*) i programiranje memorija na uređajima (koje su nama od većeg interesa).

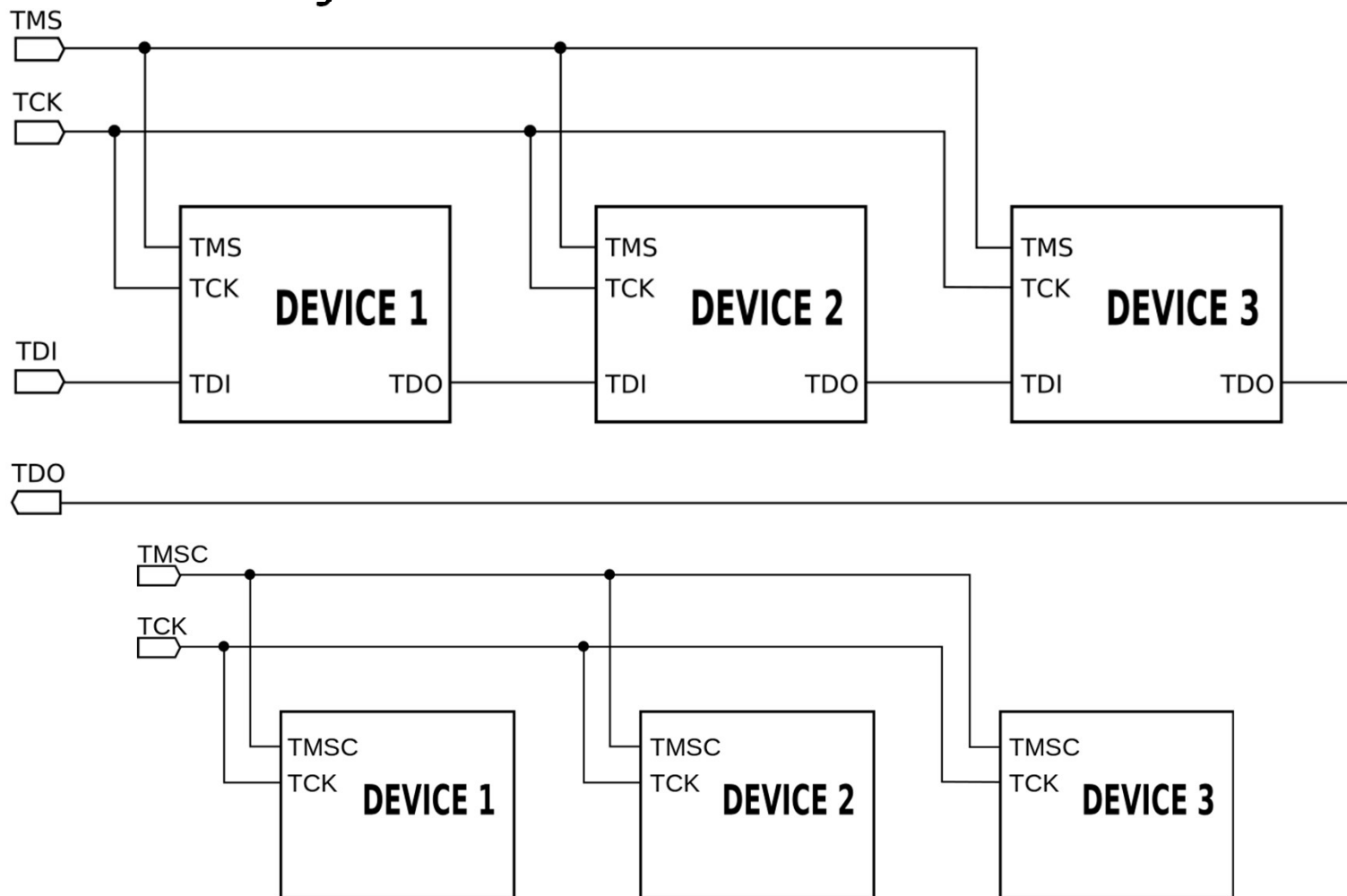
Boundary Scan

- Integrirana kola koja podržavaju Boundary Scan nam omogućavaju da ručno manipuliramo (čitamo i upisujemo vrednosti) signalima na tom integriranom kolu (najčešće pinovima, ali ponekad i internim signalima).
- Da bi se to postiglo na željene signale se dodaje *scan cell* koji se povezuju u *boundary scan shift register* (BSR) povezan na TAP.

Fizički interfejs

- Tri podržana interfejsa sa 2, 4 ili 5 linija.
- Serijski protokol koji omogućava ulančavanje više uređaja.
- Linije:
 - **TDI** (Test Data In)
 - **TDO** (Test Data Out)
 - **TCK** (Test Clock) 10-100 MHz (zavisi od najsporijeg uređaja u lancu)
 - **TMS** (Test Mode Select) prolazi kroz automat stanja JTAG protokola
 - **TRST** (Test Reset) opcion

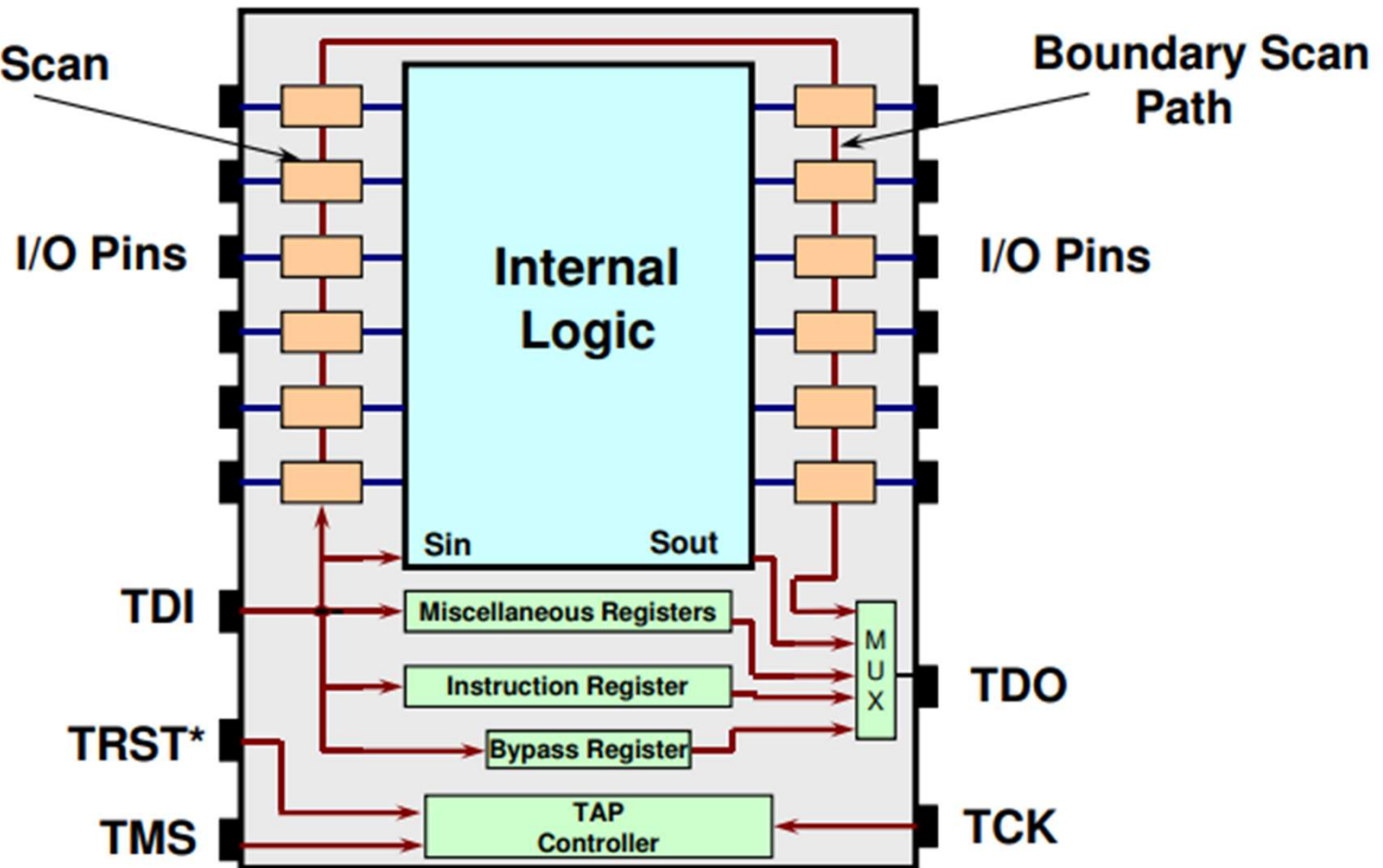
Fizički interfejs



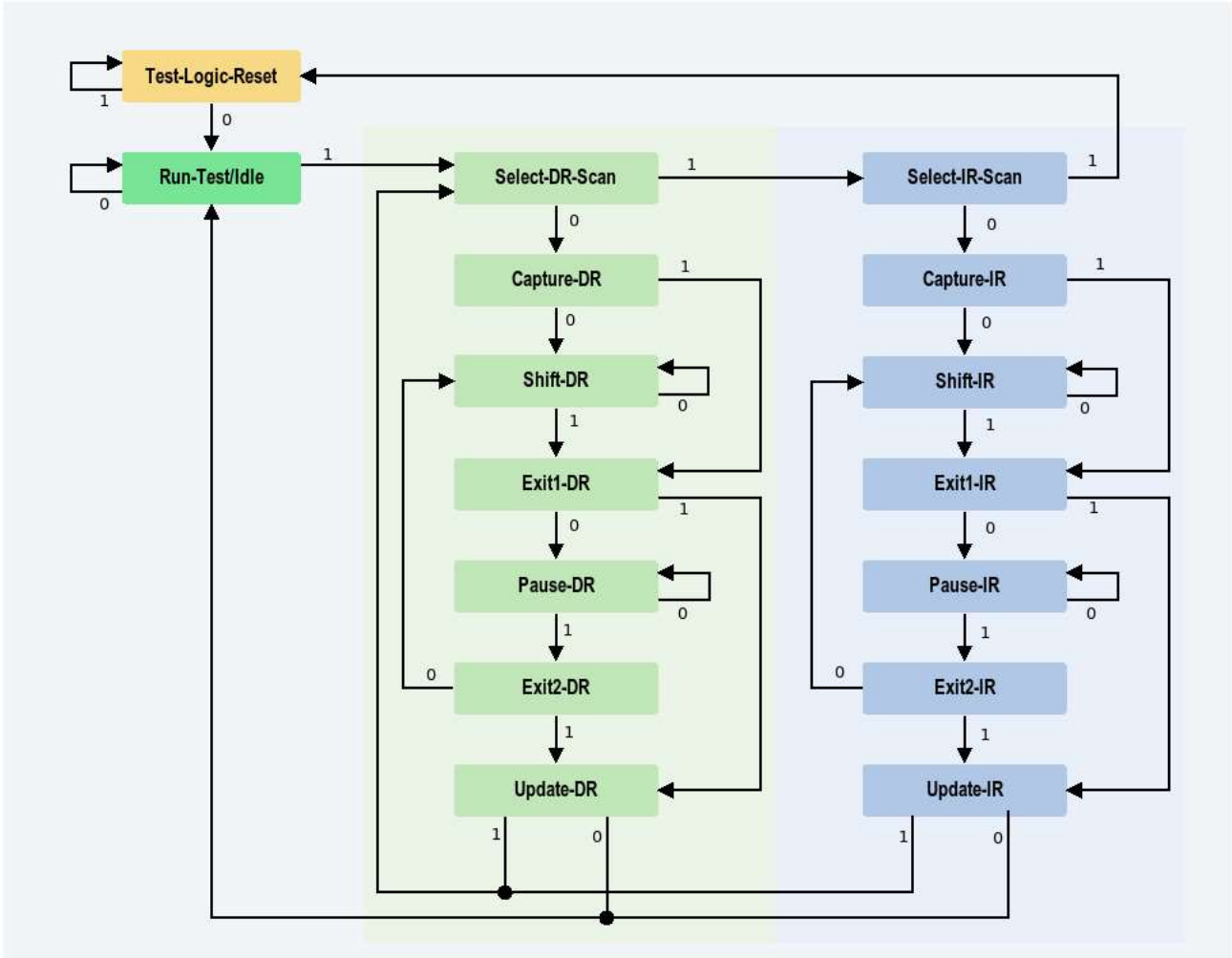
Protokol

- Fizički interfejs povezuje jedan ili više TAP-ova (može postojati i više TAP-ova na jednom čipu) sa JTAG adapterom koji njima upravlja korišćenjem JTAG protokola.
- JTAG protokol je baziran na automatu stanja sa 16 stanja, svaki takt prelazi se u stanje u zavisnosti od nivoa **TMS** linije.
- Automat stanja zajedno sa **TDI** i **TDO** linijama omogućava upisivanje i čitanje instrukcijskog registra (IR) i registra sa podacima (DR).
- IR je fiksne veličine koju specifikuje proizvođač uređaja, dok veličina DR-a može zavisiti od trenutne instrukcije (i takođe je specifikuje proizvođač uređaja).

JTAG implementacija



JTAG automat stanja



Standardne instrukcije

- **BYPASS** (sve jedinice), len(DR)=1, omogućava da se TAP prebaci u bypass mod u kome ignoriše instrukcije.
- **IDCODE**, len(DR)=32, dohvata identifikator uređaja.
- **EXTEST, PRELOAD, SAMPLE**
- Opcione instrukcije:
- **CLAMP, HIGHZ, INTEST, RUNBIST, USERCODE**
- Dodatne instrukcije se definišu u BSDL fajlu.

In Circuit Debugging

- Proizvođači mikrokontrolera i drugih programabilnih čipova, su ubrzo uvideli da je relativno lako proširiti JTAG dodavanjem novih instrukcija, kako bi omogućili komunikaciju sa modulima za debugovanje na samom čipu.
- Ovakav način debugovanja je podržan na skoro svim arhitekturama koje su danas u upotrebi, kroz različite vlasničke protokole.
- Neki od primera su: ARM CoreSight, Intel Processor Trace...
- Sami moduli za debugovanje omogućavaju većinu funkcionalnosti na koje smo navikli pri debugovanju softvera na PC računarima.

ARM CoreSight

Table 222. 32-bit debug port registers addressed through the shifted value A[3:2]

Address	A[3:2] value	Description
0x0	00	Reserved, must be kept at reset value.
0x4	01	DP CTRL/STAT register. Used to: <ul style="list-style-type: none"> Request a system or debug power-up Configure the transfer operation for AP accesses Control the pushed compare and pushed verify operations. Read some status flags (overrun, power-up acknowledgements)
0x8	10	DP SELECT register: Used to select the current access port and the active 4-words register window. <ul style="list-style-type: none"> Bits 31:24: APSEL: select the current AP Bits 23:8: reserved Bits 7:4: APBANKSEL: select the active 4-words register window on the current AP Bits 3:0: reserved
0xC	11	DP RDBUFF register: Used to allow the debugger to get the final result after a sequence of operations (without requesting new JTAG-DP operation)

Table 228. Core debug registers

Register	Description
DHCSR	The 32-bit Debug Halting Control and Status register This provides status information about the state of the processor enable core debug halt and step the processor
DCRSR	The 17-bit Debug Core register Selector register: This selects the processor register to transfer data to or from.
DCRDR	The 32-bit Debug Core register Data register: This holds data for reading and writing registers to and from the processor selected by the DCRSR (Selector) register.
DEMCR	The 32-bit Debug Exception and Monitor Control register: This provides Vector Catching and Debug Monitor Control. This register contains a bit named TRCENA which enable the use of a TRACE.

Table 221. JTAG debug port data registers

IR(3:0)	Data register	Details
1111	BYPASS [1 bit]	-
1110	IDCODE [32 bits]	ID CODE 0x3BA00477 (Arm® Cortex®-M3 r1p1-01rel0 ID Code)
1010	DPACC [35 bits]	Debug port access register This initiates a debug port and allows access to a debug port register. <ul style="list-style-type: none"> When transferring data IN: <ul style="list-style-type: none"> Bits 34:3 = DATA[31:0] = 32-bit data to transfer for a write request Bits 2:1 = A[3:2] = 2-bit address of a debug port register. Bit 0 = RnW = Read request (1) or write request (0). When transferring data OUT: <ul style="list-style-type: none"> Bits 34:3 = DATA[31:0] = 32-bit data which is read following a read request Bits 2:0 = ACK[2:0] = 3-bit Acknowledge: <ul style="list-style-type: none"> 010 = OK/FAULT 001 = WAIT OTHER = reserved Refer to Table 222 for a description of the A[3:2] bits
1011	APACC [35 bits]	Access port access register Initiates an access port and allows access to an access port register. <ul style="list-style-type: none"> When transferring data IN: <ul style="list-style-type: none"> Bits 34:3 = DATA[31:0] = 32-bit data to shift in for a write request Bits 2:1 = A[3:2] = 2-bit address (sub-address AP registers). Bit 0 = RnW = Read request (1) or write request (0). When transferring data OUT: <ul style="list-style-type: none"> Bits 34:3 = DATA[31:0] = 32-bit data which is read following a read request Bits 2:0 = ACK[2:0] = 3-bit Acknowledge: <ul style="list-style-type: none"> 010 = OK/FAULT 001 = WAIT OTHER = reserved There are many AP registers (see AHB-AP) addressed as the combination of: <ul style="list-style-type: none"> The shifted value A[3:2] The current value of the DP SELECT register
1000	ABORT [35 bits]	Abort register <ul style="list-style-type: none"> Bits 31:1 = Reserved Bit 0 = DAPABORT: write 1 to generate a DAP abort.

Programiranje memorija

- JTAG se takođe može proširiti mogućnošću da učitava sadržaj memorije u RAM (kod FPGA čipova) ili FLASH (kod mikrokontrolera).
- Ova funkcionalnost se može realizovati na nekoliko načina:
 - Modul za debugovanje zadaje instrukcije procesoru koje vrše upis u memoriju
 - Kontroler memorije ili sam memorijski čip ima svoj TAP
 - Modul za debugovanje omogućava direktan pristup magistrali u funkciji gazde.
- Neki JTAG adapteri imaju mogućnost da emuliraju memoriju te mogu značajno ubrzati ciklus debugovanja.

JTAG adapteri

- JTAG adapteri konvertuju fizički JTAG interfejs u neki interfejs koji host računar razume (najčešće USB ili Ethernet).
- Sami adapteri mogu biti relativno prosti uređaji ili mogu implemetirati neke funkcionalnosti direktno unutar adaptera.
- Glavna kompleksnost ovih adaptera leži u softveru koji upravlja samim adapterom, ti softveri su često specijalizovani za jednu arhitekturu ili vrstu uređaja i/ili su veoma skupi.

JTAG adapteri



Segger J-Link



XJTAG XJLink2



ALTERA USB-Blaster 2

Ekstenzije i alternative

- Iako je JTAG poprilično rasprostranjen standard, proizvođači često pored JTAG-a omogućavaju slične funkcionalnosti kroz vlasničke protokole ili vlasničke ekstenzije na JTAG standard.
- Jedna od najpoznatijih alternativa je ARM Single Wire Debug koji je u velikoj mjeri sličan JTAG interfejsu sa dve linije.
- Za potrebe debugovanja modernih Core procesora, Intel je razvio ekstenziju JTAG standarda koja koristi čak 60 pinova.