

Politechnika Śląska w Gliwicach
Wydział Automatyki Elektroniki i Informatyki

Wykorzystanie standardu JTAG do programowania i debugowania układów logicznych

Promotor

dr inż. Jacek Loska

Wojciech Klimeczko

1.Co to jest JTAG

JTAG (IEEE 1149.1)

W 1985 roku powstaje organizacja pod nazwą Join Test Action Group stowarzyszająca około 200 producentów układów elektronicznych (JTAG).

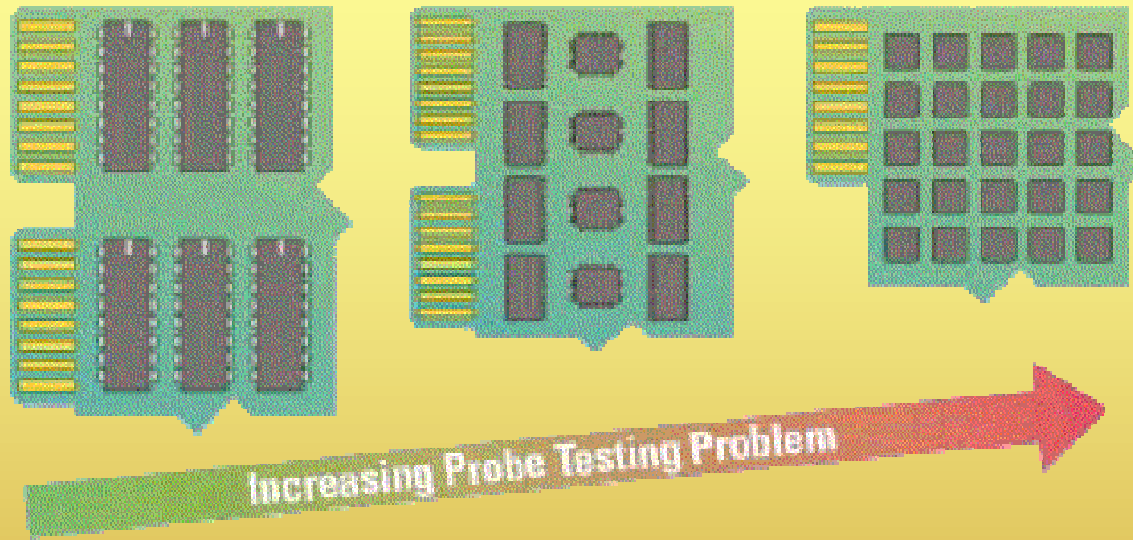
W 1990 roku rozwiązania wprowadzone do produkcji układów elektronicznych przez stowarzyszenie JTAG zostają uznane za międzynarodowy standard pod nazwą IEEE Std 1149.1 (Institute of **E**lectrical and **E**lectronics **E**ngineers)

IEEE 1149.1 – Test Access Port and Boundary-Scan Architecture

Boundary Scan – pozwala na kompletne sterowanie oraz obserwowalność wyprowadzeń układu scalonego zgodnego z JTAG realizowane za pomocą oprogramowania (**bez potrzeby podłączania urządzeń testujących !!!**)

Układy pracują w ich normalnym trybie lub w sposób umożliwiający analizę wejść oraz ustawianie stanów wyjść

2. Po co powstał JTAG



1. Każdy układ produkowany przez firmę elektroniczną musi zostać poddany testom przed wprowadzeniem do sprzedaży.
2. Wraz z rozbudową układów elektronicznych pojawią się problemy w przeprowadzeniu testów z takim układami.

Zalety JTAG

- Ustawianie oraz obserwacja stanów bez fizycznego połączenia
- Testowe sygnały mogą być generowane nie tylko na wejściu płytki ale również w dowolnym punkcie wewnętrznym bez potrzeby przerywania połączeń
- Redukcja punktów testowych na płytkach
- Automatyczna identyfikacja układów
- Zauważono że standard IEEE 1149.1 (JTAG) może być wykorzystany nie tylko na etapie testowania urządzeń, ale również w czasie ich wykorzystania. Wtedy magistrala JTAG używana jest do sterowania urządzeniem i monitoringu parametrów jego pracy.
- Standard umożliwiający współpracy urządzeń różnych producentów

Wady JTAG

- Zwiększona powierzchnia układu potrzebna na realizację obwodu Boundary Scan
- Potrzebne dodatkowe wyprowadzenia
- Dodatkowy koszt procesu projektowania
- Zmniejszenie szybkości działania układu przez wprowadzone dodatkowe bloki
- Zwiększony pobór mocy

**Ilość bramek potrzebnych do zaimplementowania Boundary Scan Test
dla układu z 10000 bramek i 40 wyprowadzeń**

Rejestry Boundary Scan	680
Kontroler TAP	131
Rejestr instrukcji	28
Rejestr Bypass	9
Pozostałe	20
RAZEM	868

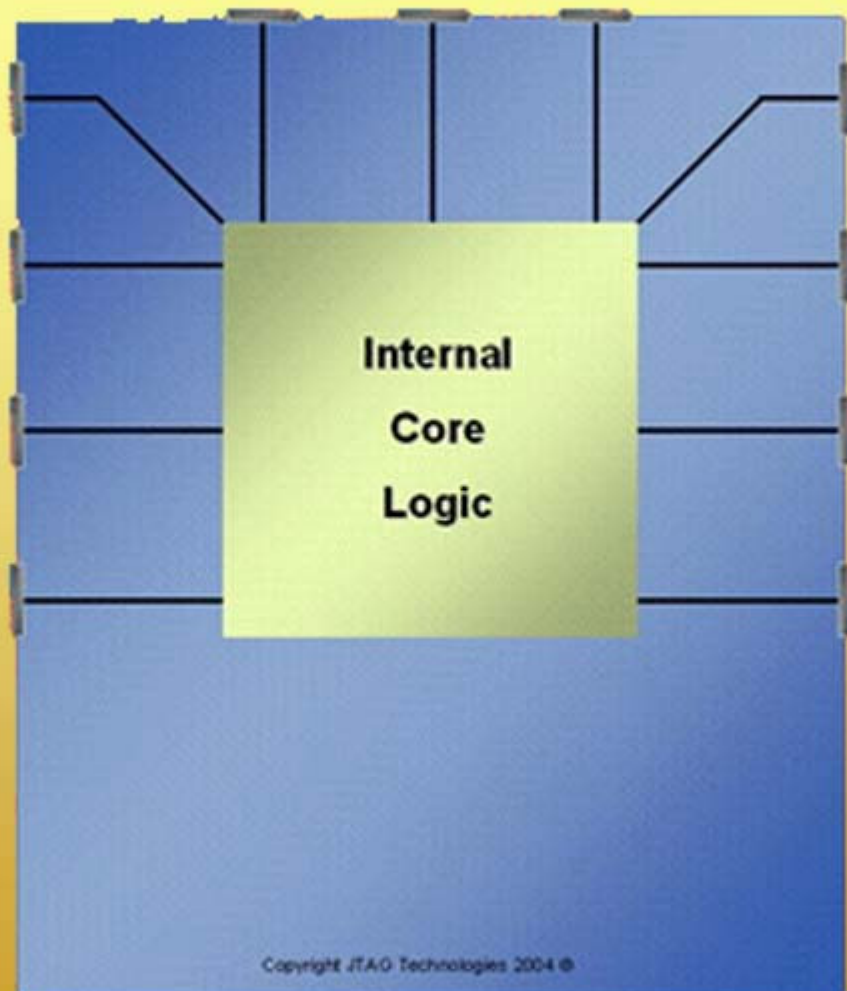
3 .Budowa JTAG

Obwód magistrali diagnostycznej:

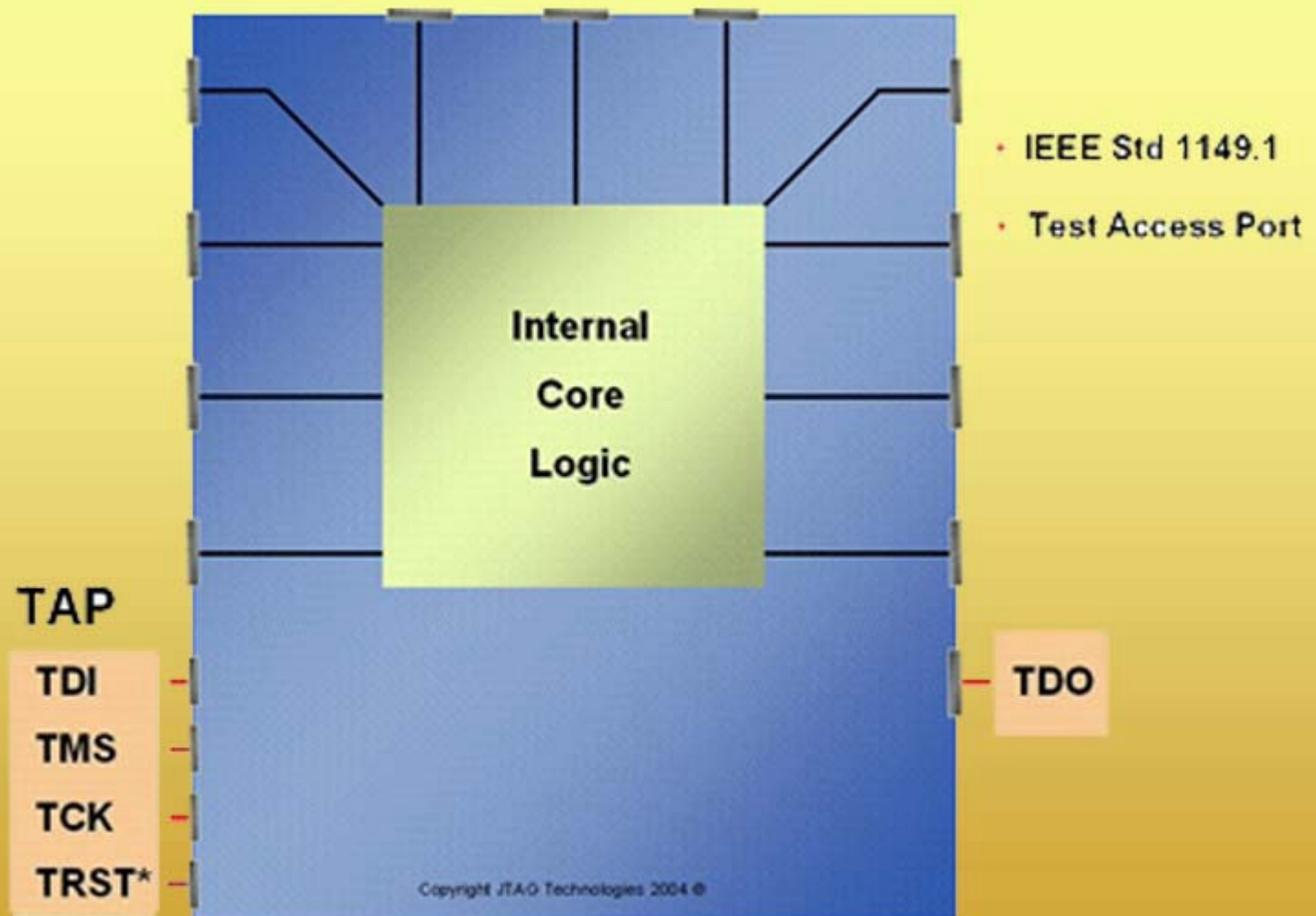
1. Krawędziowe rejestry diagnostyczne
2. Rejestry ominięcia
3. Rejestr instrukcji
4. Opcjonalne rejestry specjalne
5. Układ zarządzający TAP

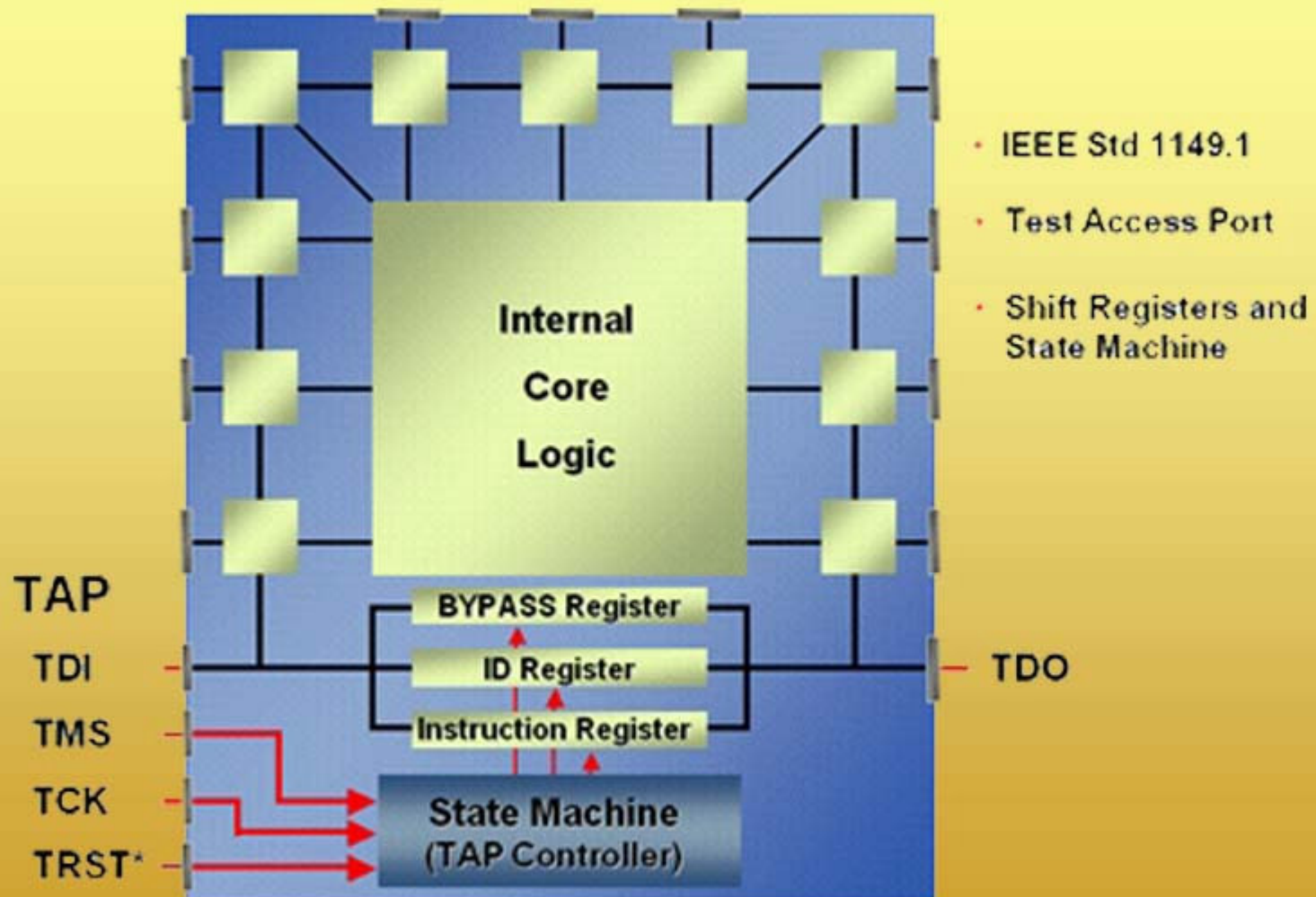
W skład magistrali krawędziowej wchodzi także 4 linie:

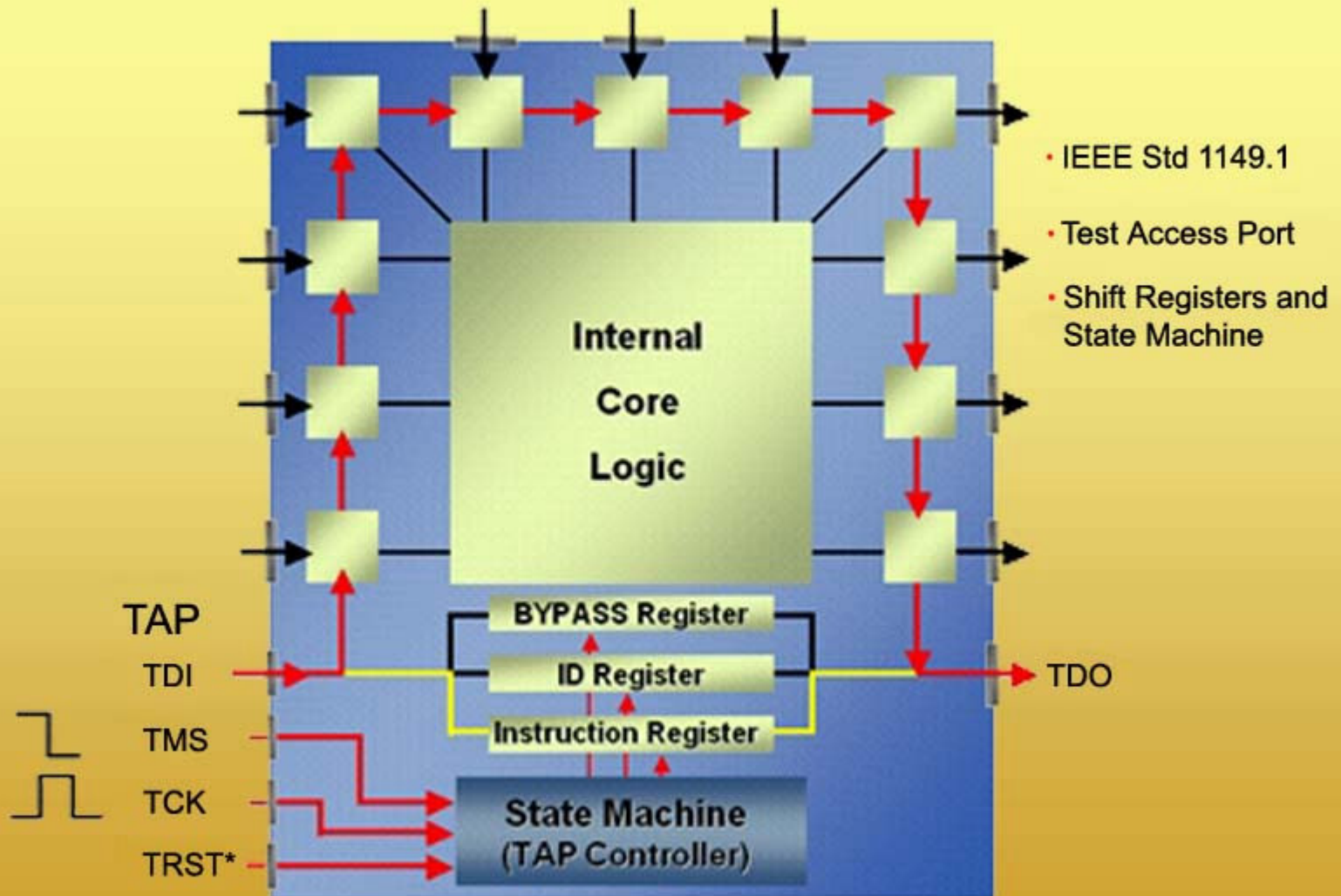
1. TCK – Wejście sygnału zegarowego, taktującego automat TAP oraz rejestr instrukcji.
2. TMS – Wejście sterujące pracą automatu TAP.
3. TDI – Szeregowe wejście danych do konfiguracji i testowania.
4. TDO – Szeregowe wyjście danych wyprowadzanych z rejestru BST.
5. TRST – Wejście sygnału zerującego TAP i inne elementy BST.

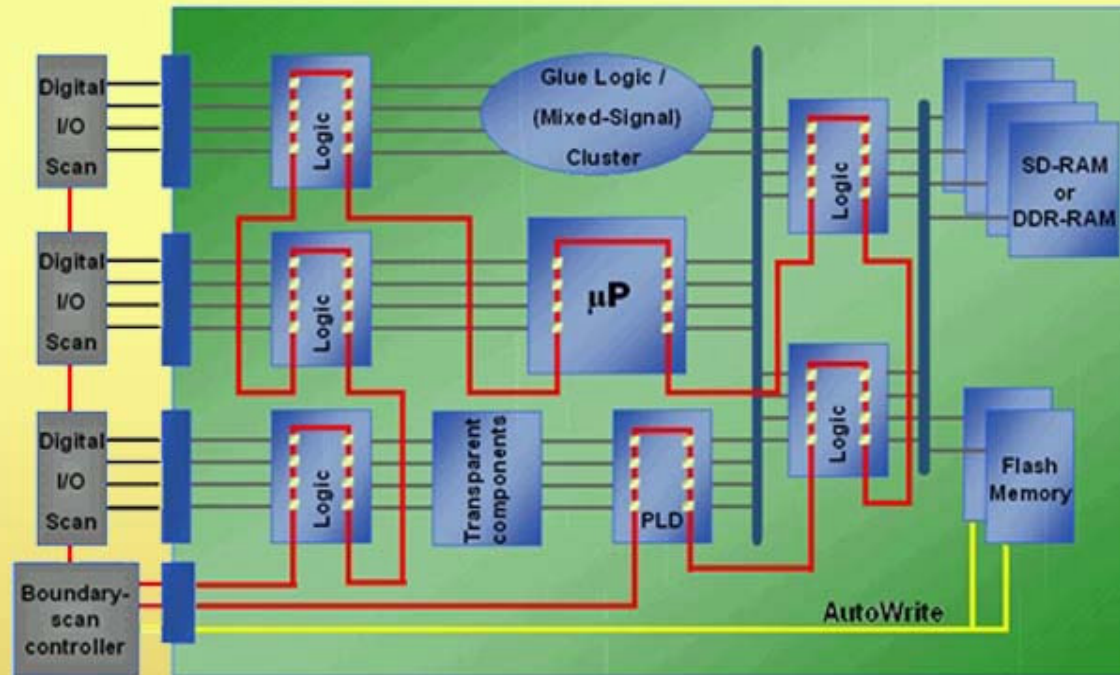


Copyright JTAG Technologies 2004 ©









Co możemy testować:

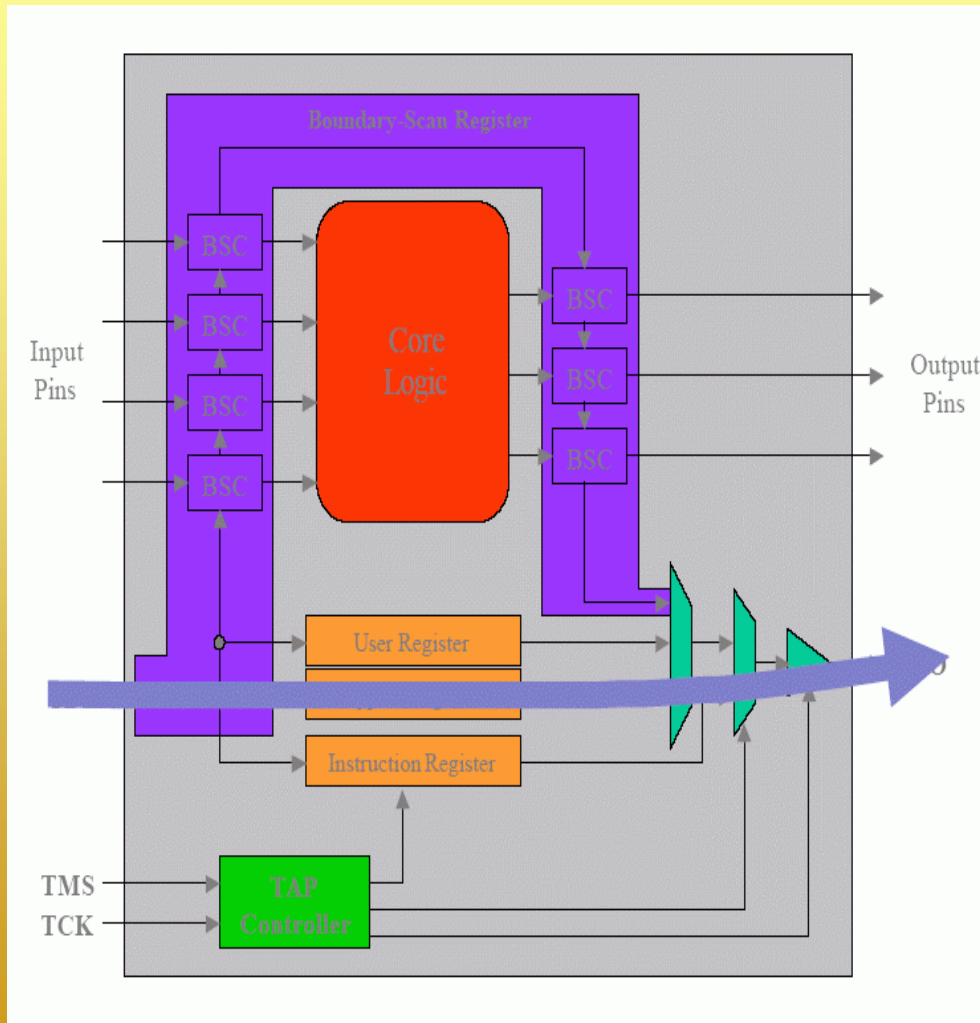
- połączenie między elementami
- połączenia z elementami nie zawierającymi obwodów testowych
- połączenia z pamięciami RAM, Flash
- połączenia z wejściami układu

Twórcy interfejsu JTAG założyli, że nie ma potrzeby szczegółowego testowania wewnętrznych fragmentów układów, o których poprawną pracę powinien zadbać projektant na etapie projektowania struktury logicznej.

4. Rozkazy JTAG

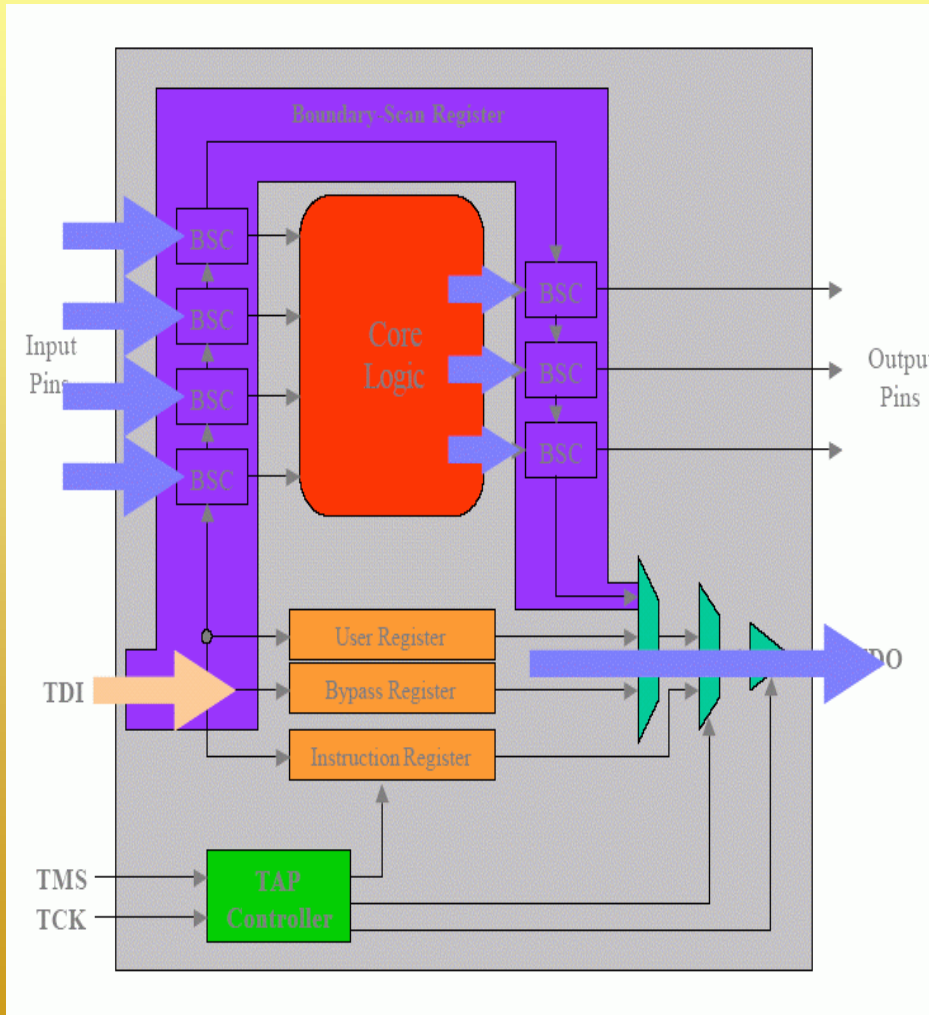
- 1149.1 definiuje 9 instrukcji, 3 są obowiązkowe
- Instrukcje obowiązkowe: BYPASS, SAMPLE/PRELOAD i EXTEST
- instrukcje nieobowiązkowe: INTEST, RUNBIST, CLAMP, HIGHZ, IDCODE oraz USERCODE
- Kody instrukcji i ich długość definiowane przez producenta układu (minimum 2 bity długości instrukcji)
- Producent układu może zdefiniować dowolną liczbę specyficznych instrukcji oraz zdefiniować ich działanie

Instrukcja BYPASS (wymagany)



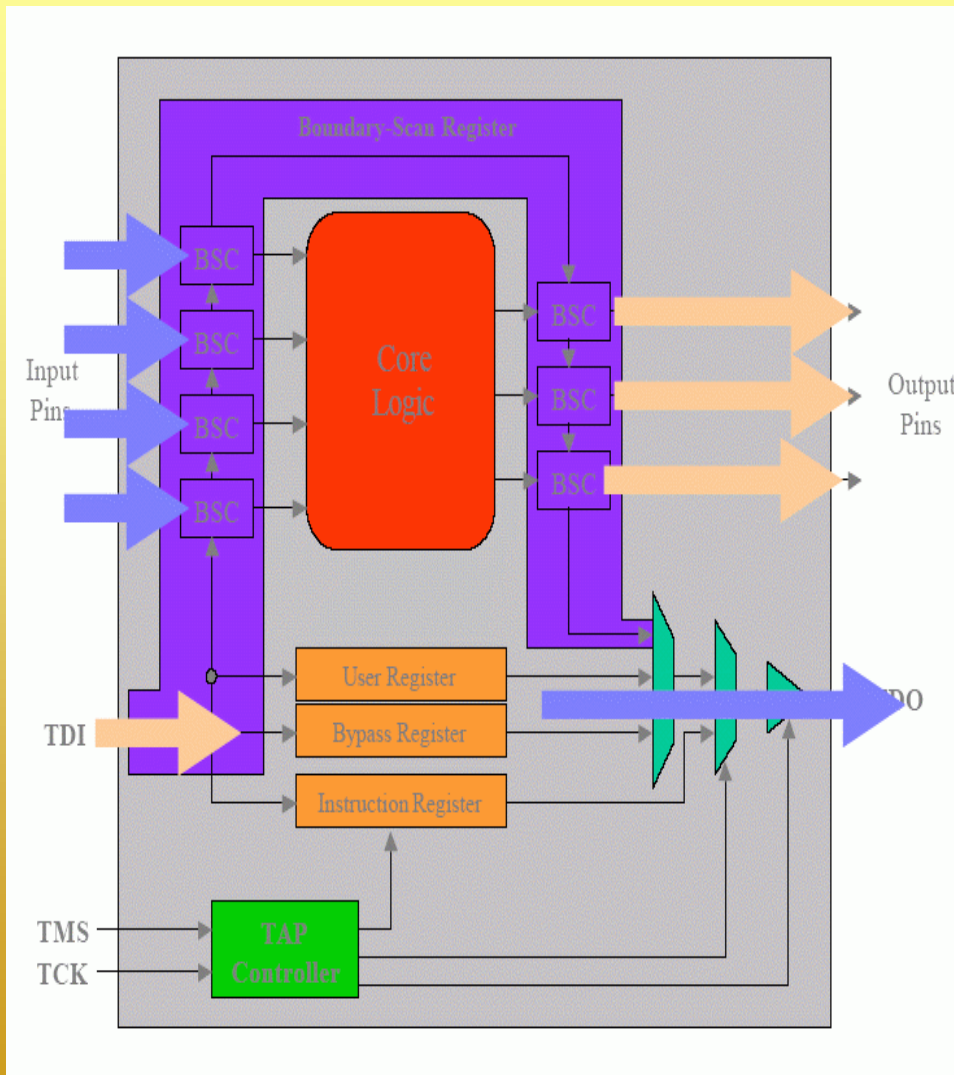
- Układ pozostaje w normalnym trybie pracy
- Jednobitowy rejestr między TDI a TDO
- Dane są przesyłane przez układ bez zakłócania pracy
- Nie jest przerywany łańcuch JTAG
- Wysłanie do układu nieznanego rozkazu wprowadza go w stan „bypass”

Instrukcja SAMPLE/PRELOAD (wymagany)



- Układ pozostaje w normalnym trybie pracy
- Boundary-Scan Register wpięty między TDI oraz TDO
- Pobiera funkcjonalne dane wchodzące i wychodzące z układu
- Dane z Boundary-Scan Register dostępne podczas operacji skanowania (zawierające wejścia i wyjścia układu)
- Ładuje do BSC dane wykorzystywane przez rozkaz EXTEST – ładuje dane przed wejściem do trybu testowania

Instrukcja EXTEST (wymagany)



- Układ w zewnętrznym trybie testowym
- Boundary-Scan Register wpięty między TDI oraz TDO
- BSR steruje danymi wychodzącymi z układu
- Przed operacją skanowania dane wejściowe zapisywane do BSR
- Umożliwia np. sprawdzanie połączeń między układami

PROGRAMOWANIE UKŁADÓW LOGICZNYCH

1. Klasyczne metody programowania

Pierwszymi układami programowalnymi, jakie pojawiły się na rynku, były jednokrotnie programowalne pamięci PROM.

W skutek rozwoju technologii układów półprzewodnikowych powstały matryce pamięciowe EPROM, które można programować wielokrotnie. Kolejnym krokiem w rozwoju układów programowalnych było powstanie pamięci EEPROM.

Do konfigurowania (programowania) układów PLD niezbędne były specjalne programatory. Szybki rozwój układów powodował, że użytkownicy byli zmuszeni do inwestowania w coraz nowszy, kosztowny i mało uniwersalny sprzęt.

Stopniowa unifikacja algorytmów i parametrów programowania układów pochodzących od różnych producentów umożliwiła powstanie programatorów uniwersalnych.

Obecne programatory obsługują co najmniej kilka tysięcy różnych układów.



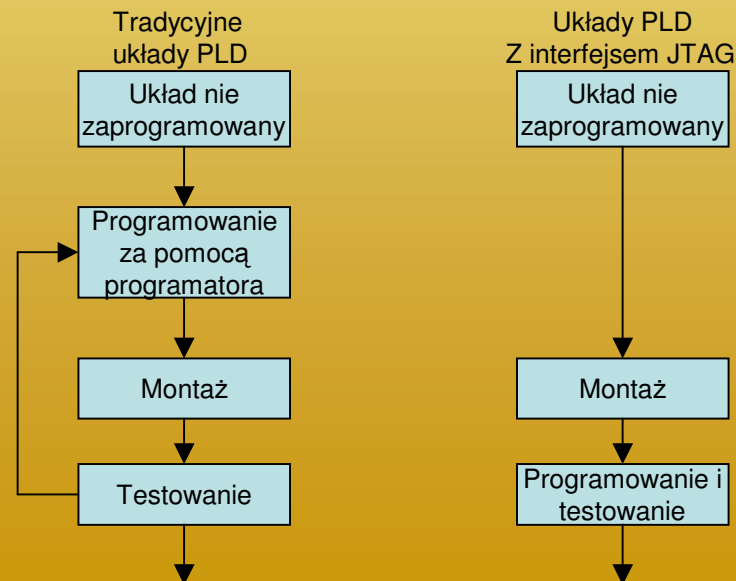
Pamięci SRAM

Komórki pamięci w niektórych układach FPGA i niektórych CPLC o dużej gęstości upakowania wykonane są w technologii SRAM.

Takie układy wymagają konfiguracji po każdym jego włączeniu. Do tego celu służą wbudowane w układy FPGA interfejsy, których zadaniem jest przepisanie zawartości zewnętrznej pamięci PROM, EPROM, EEPROM do wewnętrznej pamięci konfiguracji.

Metody konfiguracji układów PLD, za pomocą zewnętrznej pamięci, jest ogromną niedogodnością podczas prac konstrukcyjnych związanych z wielokrotnym przeprogramowywaniem pamięci nieulotnych.

Znacznie lepszą metodą jest bezpośrednio modyfikowanie zawartości SRAM za pomocą sprzętowego interfejsu z pominięciem nieulotnej pamięci konfiguracyjnej.



Programowanie przez JTAG

Standard JTAG pierwotnie, miał służyć do testowania złożonych modułów cyfrowych po ich zmontowaniu na płytkach drukowanych.

Pierwszym rozszerzeniem funkcjonalnym, dodanym do podstawowego zestawu funkcji JTAG-a, było umożliwienie programowania pamięci EEPROM lub Flash. Wraz z pojawieniem się układów z pamięciami SRAM, po raz kolejny rozszerzono specyfikację standardu.

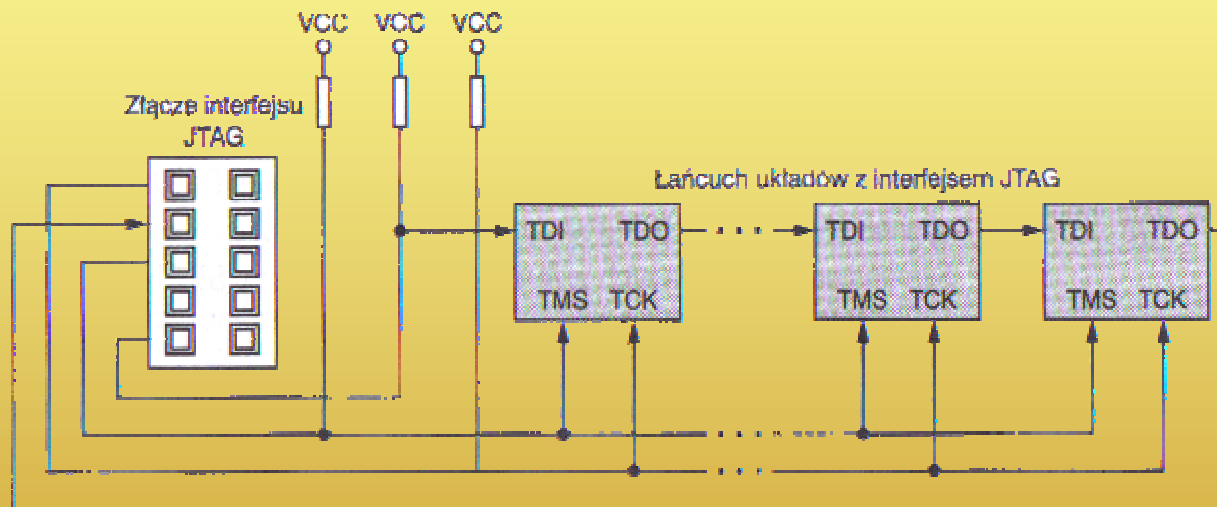
Obydwa procesy czyli testowanie i programowanie układów z interfejsem JTAG przebiegają w podobny sposób.

Najważniejszą różnicą jest to iż przy testowaniu wykorzystuje się rejestry ścieżki krawędziowej, a podczas programowania rejestry ISP/ICR. Rejestry te są opcjonalnym rozszerzeniem struktury interfejsu.

Do opisanych wcześniej 3 obowiązkowych instrukcji, w przypadku układów programowalnych dochodzą jeszcze dwie:

- IDCODE – umożliwia odczytanie sygnatury układu.
- USERCODE – umożliwia odczytanie sygnatury zapisanego projektu.

Ponieważ w opisie standardu JTAG nie uwzględniono rejestrów ISP/ICR, każdy producent układów programowalnych stosuje własne zestawy „ściśle dopasowane do układu (brak wpływu na kompatybilność standaryzowanych fragmentów układu, przeznaczonych do testowania).



W przypadku programowania wielu układów połączonych szeregowo, programowany jest kolejno każdy układ znajdujący się w szeregu.

Dzięki możliwości implementacji własnych rozwiązań, niektórzy producenci układów zaimplementowali możliwość jednoczesnego programowania wszystkich układów znajdujących się w łańcuchu.

Po co programować przez JTAG.

Wszelkie urządzenia, układy dostępne na rynku są wyposażone w złącza transmisji szeregowej np. RS232 czy USB, umożliwiające ich programowanie.