TEST SOFTWARE FOR EMBEDDED PC

Karel MASAŘÍK, Master Degree Programme (5)
Dept. of Computer Systems, FIT, BUT
E-mail: xmasar01@stud.fit.vutbr.cz

Supervised by: Ing. Martin Drahanský

ABSTRACT

New technologies, which offer the production of some electronic devices, produce permanent smaller and smaller components. These components are concentrated on very small areas. This kind of miniaturization brings always smaller connections among these components, which are difficult accessible and can not be measured. In the last years, the lines between the components were based on certain measurement points, using different test methods (e.g. a bed of the nails method). These outdated measurement methods can not be used now. Nowadays, instead these methods, the JTag technology is used or the components are tested simply by a firmware. But software test can be used only at the embedded PC. The reason for it is that most important components (e.g. boat memory, etc.) are considered as to be without errors. For the JTag-test is necessary to have at least one command component with the TAP controller and all test components could be then tested. The JTag- test interface is more universal as the software one and is intended not only for the embedded PCs, but it can be used for test of automatons and simple devices. The price of the components with the TAP controller is nearly comparable with the price of the components without the TAP controller.

1 ÚVOD

Cílem této práce je vytvořit testovací software pro zařízení založené na PC platformě. Testovací software tvoří část mé diplomové práce a je vytvářen ve spolupráci s německou firmu ATRON. V koncepci testovacího softwaru kladla firma hlavní požadavek na universálnost testeru, tzn. možnost testovat všechny výrobky, označme je jako platformu X, vytvářené touto firmou. Platformu X tvoří jízdní automaty, mobilní zařízení a jízdní kompakty (FR kompakty, tj. vlastní embedded PC). Každé z těchto zařízení má jedinečnou architekturu, která poskytuje jen omezenou množinu testovacích rozhraní. Většina platforem obsahuje součástky s JTagem, na některých platformách jsou dostupné měřící body, jež je možné pomocí měřících karet měřit, FR kompakty obsahují navíc testovací firmware. Každé testovací rozhraní je dostupné řídícímu PC (běží na něm tester) přes vlastní komunikační kanál. Koncept softwaru zahrnoval požadavek na uložení všech testovací výsledků do firemní databáze. Průběh a výsledky testů by měli být zobrazovány uživateli, který výrobek testuje. Testovací software má být nasazen ve třech oblastech: celková montáž produktu, servis (v domácnosti, u zákazníka) a poskytovat prostředí pro vývoj testů cílových zařízení.

2 POPIS ARCHITEKTURY TESTOVACÍHO SOTWARE

Testovací software je rozdělen na dvě části: *server* a *tester*. Každá z těchto částí bude popsána níže. Obě části pracují s databází (správa uživatelů a uložení výsledků testů), pro komunikaci s uživateli používají dialogové boxy.

3 SERVER

Server slouží k přihlašovaní a zaznamenávání činnosti obsluhy testeru, a především k vlastnímu spuštění testeru. Tester je spuštěn po úspěšném přihlášení (autorizaci) uživatele. Bez autorizace nemá uživatel přístup k testeru, k vlastní databázi s výsledky předešlých testů a nemůže tato data zdiskreditovat. Při přihlašování udává uživatel své jméno, heslo a testovací mód, v němž má tester běžet. Testovací módy jsou tři: *produkční*, *servisní* a *vývojový*. Na základě testovacího módu je ovlivněno chování testeru. Server je plně konfigurovatelný pomocí XML souborů.

4 TESTER

Tester slouží k provádění vlastních testů. Architektura testeru je rozdělena na tři nezávislé části: *testovací rozhraní*, *testovací knihovny* a vlastní *testovací aplikace*. Vývoj jednotlivých částí je možné provádět nezávisle na ostatních částech. Testovací rozhraní a testovací knihovny jsou uloženy v šifrovaných dynamických knihovnách, jež jsou použity testovací aplikací.

4.1 TESTOVACÍ ROZHRANÍ

V současnosti využívá tester k realizaci testů dva druhy testovacích rozhraní. První tvoří softwarové rozhraní, jež komunikuje s testovacím firmwarem běžícím na embedded PC. Druhé je JTag, jež samo provádí testy. V budoucnu přibude rozhraní měřících karet.

JTag testovací rozhraní

Rozhraní využívá pro komunikací s cílovým zařízení paralelní port. Architektura a chování každé JTag komponenty je uložena v souboru, jež obsahuje informace z BSDL (Boundary-Scan Description Language) souboru. Za pomoci těchto souborů jsou inicializovány všechny JTag (řídící) periferie na cílovém systému. Při inicializaci JTag rozhraní zjišťuje tester počet a pořadí řídících periferií. Pomocí řídící periferie (většinou se jedná o procesor nebo FPGA) jsou dostupné ostatní testované součástky cílového systému. Takto jsou pak realizovatelné testy např. ROM/RAM pamětí, systémové sběrnice nebo komunikačních portů. Toto rozhraní je tedy použito v případě, že softwarové rozpoznání testovacího rozhraní selhalo z důvodu chyby, např. komunikačního portu nebo chybné bootovací paměti. JTag testy periferií jsou popsány pomocí abstraktních příkazů, instrukcí a abstraktních sběrnic. Pomocí abstraktních sběrnic je propojena testovací periferie s řídící. Jedná se o abstrakci, pomocí které je určitý shluk vodičů (řídící signály, datová sběrnice) viděn jako souvislý celek, tzn. je viděn jako registr, do kterého je možné zapisovat nebo je možné z něj číst. Jednotlivé bity tohoto registru pak tvoří logické hodnoty na jednotlivých vodičích. Abstraktní instrukce pak popisují posloupnosti, ve kterých jsou tyto registry nastavovány, popisují tedy posloupnost logických hodnot na jednotlivých vodičích testované periferie. Hodnoty registrů jednotlivých posloupností instrukcí jsou nastavovány pomocí abstraktních příkazů. Abstraktní příkazy shlukují abstraktní instrukce do sekvencí, jejichž realizací je možné nastavit testovanou součástku do určitého stavu. Takto jsou například

programovány Flash paměti, nebo je možné číst a zapisovat do SDRAM paměti. Jednotlivé příkazy definují takzvanou master řídící periferii (musí ji definovat každé testované zařízení) a volitelně slave řídící periferii. Master řídící periferie provádí většinou zápis dat do sběrnice (registru), slave periferie má vždy opačný tok dat než master. Tedy pokud master zapisuje data, slave z ní musí číst. Pokud master čte, musí nutně slave data zapisovat. Testovaná součástka může definovat i více slave periferií (např. více FPGA). Pak je tok dat mezi masterem a několika slave periferiemi podobný broadcastu v počítačové síti. Tento prvek rozhraní dovoluje testovat např. zkrat mezi vodiči systémové sběrnice cílového systému (ISA apod.). Všechny abstraktní prvky (příkazy, instrukce, atd.) jsou definovány v XML souborech, je tedy možné velmi rychle vytvořit nový test testované periferie. Realizace testů je podmíněna funkčností JTag komunikačního kanálu (cca. 5 vodičů) a funkčností všech řídících periferií cílového systému (většinou jsou sériově propojeny přes Boundary Scan Registr).

Softwarové testovací rozhraní

Softwarové testovací rozhraní umožňuje testovat periferie pomocí firmwaru, jež běží v embedded PC. Rozhraní komunikuje s firmwarem pomocí vlastního protokolu přes sériový port. Protokol přenáší příkazy pro spuštění testu, zjištění stavu testu a výsledku testu. Součástí firmwaru je tzv. SW (software) Monitor. Jedná se o program, jež je spuštěn po restartu embedded systému. SW Monitor realizuje testy periferie pomocí atomických testovacích funkcí, které jsou identifikovány pomocí jedinečného čísla a jsou uloženy v paměti embedded PC. Atomické funkce provádějí nejčastěji sekvence zápisů a čtení do/z registrů periferie za účelem zjištění poruchy. Výsledek testu pak předávají SW Monitoru. Funkčnost testovacího rozhraní závisí na funkčnosti cílového systému, tzn. zda je možné komunikovat se SW monitorem (potřebujeme více provozuschopných součástek než u JTag rozhraní). Pomocí tohoto rozhraní jsou většinou prováděny testy nesystémových periferií (tiskárna či displej).

4.2 TESTOVACÍ KNIHOVNY

Každé testované zařízení vlastní jednu knihovnu. V testovací knihovně jsou uloženy testovací rutiny. Testovací rutina tvoří propojovací článek mezi aplikací testeru a testovacím rozhraním. Aplikace ji parametrizuje parametry testu a aktuálně použitým testovacím rozhraním. Vlastní rutina tato data po zpracování předává testovacímu rozhraní. Zpracování dat může znamenat změnu či doplnění hodnot uživatelem za pomocí dialogových boxů. V případě JTag testů je v těle rutiny nadefinována vlastní atomická testovací funkce periferie, za použití abstraktních příkazů. Funkčnost JTag atomické funkce je podobná SW funkci.

4.3 TESTOVACÍ APLIKACE

Tester tvoří rozhraní mezi uživatelem, jež test provádí, a cílovým testovaným systémem. Tester komunikuje s uživatelem pomocí jednoduchého dialogového boxu, ve kterém může uživatel na základě testovacího módu testeru vybrat jeden za tří typů testovaných objektů: celý cílový systém (*produkční mód*), některé periferie (*servisní mód*) a jednotlivé testovací rutiny (*vývojový mód testu*). Každý typ testovaného objetu je specifikován pomocí vlastního typu XML souborů. Objekt typu periferie obsahuje navíc dodatečný soubor, volitelně vytvářený pro každé testovací rozhraní. Pomocí tohoto objektu je provedena parametrizace testovací rutiny. XML konfigurace testeru definuje nyní dvě testovací rozhraní, ke kterým v budoucnu přibude možnost provádět testy pomocí bezdrátového spojení (radiokarty).