



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Kódfedési vizsgálatok hardveresen támogatott programvégrehajtás követéssel

Szennai Miklós BREPMI MSC 2. évf. mérnök informatikus szak
Konzulens: Scherer Balázs, MIT
Szolgáltatásbiztos rendszertervezés szakirány
Diplomatervezés I. összefoglaló
2011/12. II. félév

A beágyazott rendszerek többsége olyan alkalmazási területekhez kötődik, ahol fontos a nagyfokú megbízhatóság, biztonság, rendelkezésre állás. Ebből fakadóan nagy hangsúlyt kell fektetni azok tesztelésére, hogy az esetleges hibákat detektálni illetve javítani lehessen. A korszerű mikrovezérlők hardveresen igen nagy megbízhatósággal rendelkeznek, így a hibák javarészt szoftveres eredetűek, s emiatt a gyártók igyekeznek minél több támogatást nyújtani a tesztelések elősegítéséhez.

A struktúra alapú tesztervezési módszereket elterjedten alkalmazzák ezeken a területeken is. Ez a tesztelendő rendszerre, mint „üvegdoboz” tekint, amelynek ismert a belső felépítése, adatszerkezete, algoritmusai, illetve legtöbbször maga a forráskód is elérhető. Ezekre az ismeretekre építve végezhetünk kódfedési vizsgálatokat, amelyek arról adnak információt, hogy egy adott teszt lefutás során konkrétan mely programrészek hajtottak végre. Ezek alapján számíthatunk különböző metrikákat (vezérlési folyamat alapú kritériumok, adatfolyam alapú kritériumok).

Hagyományosan a kódfedési vizsgálatok a tesztelendő program felműszerezését igénylik, aminek hátránya, hogy módosul az eredeti kód, valamint plusz erőforrásigényt is jelent. Célunk ezen felműszerezettség mértékének csökkentését, vagy akár teljes mértékű kiváltását elérni egy ARM Cortex-M3 magú mikrovezérlőn, kihasználva annak hardveres trace és debug támogatásait.

Az ARM által kifejlesztett CoreSight debug architektúra komponenseinek alkalmas felkonfigurálásával, illetve az ezek által generált trace adatokhoz való hozzáféréssel olyan információk birtokába juthatunk, amelyekkel real-time nyomon lehet követni adott memória- illetve regisztercímeken lévő tartalmak változását, valamint a processzor által végrehajtott utasítások szekvenciáját. Ezeket az adatokat PC-re továbbítva lehetőség nyílik egy olyan komplex debug alkalmazás kialakítására, amellyel egy adott program változó-értékeinek valós idejű megfigyelését, vagy a program forrásának kódfedettségek elemzését lehet elvégezni.

Az első félév során elkészült egy Java nyelven írt, CoreSight trace-folyam értelmező architektúra, amely által a nyers adatfolyamból kinyert információkat magasabb szintű tesztelési funkcionalitást megvalósító komponensek számára tudunk elérhetővé tenni, melyek erre építve többek között utasítás, döntés vagy feltétel lefedettség metrikákat számíthatnak a futtatott kódon. A cél egy Eclipse plug-in megvalósítása, ami ezeket a funkciókat egy jól használható fejlesztőkörnyezetbe integrálva bocsátja a fejlesztők és tesztelők rendelkezésére.