

Beágyazott rendszer számítási terhelésének mérése

Embedded system computation load measurment

DRÓTOS Dániel¹, VÁSÁRHELYI József²

PhD hallgató, egyetemi tanár

Miskolci Egyetem, 3530 Miskolc Egyetemváros
{daniel.drotos, jozsef.vasarhelyi}@uni-miskolc.hu

Abstract

The computing load of embedded systems is crucial when implementing real-time applications. Processor load can be analyzed in several ways. The application described in this paper presents a possible method for analyzing processor behavior. The paper presents a developed model capable of classifying processor load. Implementation of the model was done on an FPGA (Field Programmable Gate Array) circuit, without modifying the original embedded system. The results are presented in the period of processor overload.

Keywords: FPGA, processor load, embedded systems, real-time systems, simulation

Kivonat

Beágyazott rendszerek számítási terhelése kulcsfontosságú a valós idejű alkalmazások megvalósításakor. A processzor terhelés többféleképpen elemezhető. A jelen cikkben ismertetett alkalmazás bemutat egy lehetséges módszert a processzor viselkedésének elemzésére. A cikk bemutat egy kidolgozott modellt, amely képes osztályozni a processzor terhelést. A modell megvalósítása az eredeti beágyazott rendszer módosítása nélkül történt FPGA (Field Programmable Gate Array) áramkörön. A mérési eredményeket a processzor túlterheltségének bemutatásával ismertetjük.

Kulcsszavak: FPGA, processzor terhelés, beágyazott rendszerek, valós idejű rendszerek, szimuláció

1. BEVEZETÉS

A beágyazott rendszerek tervezése általában három technológián alapul: processzor technológián, integrált áramköri technológián és tervezési technológián [1]. A processzor technológia három csoportra osztja a processzorokat: általános célú, alkalmazásspecifikus és célorientált. Az FPGA technológiában azonban a processzoroknak új felosztása is lehet, például kemény magos processzorok és lágy magos processzorok. A lágy magos processzorokat hardverleíró nyelveken (VHDL vagy Verilog) definiálják, és az FPGA-tervbe integrálják. Ily módon a beágyazott rendszer részévé válnak. A kisméretű FPGA-kban megvalósított beágyazott rendszerek esetében fontos elemezni a processzor számítási idő kihasználtságát (terhelését) az alkalmazás végrehajtása során. Az ebben a cikkben bemutatott kutatás célja a CPU (Central Processing Unit) teljesítményének növelése egy FPGA-val megvalósított beágyazott rendszeren.

Beágyazott rendszerekben FPGA áramkörök használatkor különböző módokon lehet lágymagos processzort beilleszteni. Ennek egyik módja az előre megtervezett szellemi tulajdon (IP – intellectual property) használata. A másik módszer a regiszter transzfer szintű (RTL) tervezés esetében saját processzor használata. Az IP-k használata nem alkalmas további vizsgálatokra, mivel felépítése nem engedi a belső jelekhez történő hozzáférést. A saját tervezésű CPU használatának több hátránya is van, például egy fejlesztői környezet megvalósítása is szükséges.

A beágyazott rendszer terhelés vizsgálatának legegyszerűbb módja szoft processzor mag használata, amely lehetővé teszi a processzor jeleinek hozzáférést is. Ebben az esetben az FPGA-k ideális felületet biztosítanak a vizsgálatok megvalósítására. Ez lehetővé teszi a beágyazott rendszer fejlesztését, processzorok, perifériák, memóriák és egyéb járulékos áramkörök megvalósítását is.

A fel nem használt FPGA erőforrások rendelkezésre állnak más áramkörök megvalósítására [2]. A CPU teljesítménye növelhető további áramkörök hozzáadásával, amelyek képesek a számítási feladatokat a CPU-feladatoktól függetlenül elvégezni. A fel nem használt FPGA-erőforrásokon társprocesszorokat vagy egyéb további beágyazott rendszerfeladatokat lehet megvalósítani.

A processzor működés közbeni vizsgálatával megállapítható, milyen típusú feladatot hajt végre, azaz milyen társprocesszorra lehet szükség a CPU-nak egy adott időpontban történő tehermentesítéséhez. Az FPGA azon területén, ahol a további áramkörök megvalósulnak, adaptív újrakonfigurálást alkalmazhatunk a megfelelő feladat (társprocesszor) megvalósításához, amikor a feladat feldolgozása során erre szükség van.

A működő processzor alapú rendszerek tesztelésére különféle módszereket fejlesztettek ki, amelyek segítségével ellenőrizhetők a rendszer egyes jellemzői. A számítógép összes alkatrészének tesztelésével lehetséges a hardverhibák észlelése [3]. Ez a tesztelési módszer a hardver működését ellenőrzi és a gép rendellenes működésére következtet. Egy másik módszer rendszer tervezési fázisában célozza meg elsősorban beágyazott rendszerek terhelésének tesztelését [4]. A [4] célja a tervezett szoftver futtatásához szükséges erőforrások meghatározása volt. Zhao és társai szoftvereszközöket és egy erre a célra fejlesztett speciális rendszert használtak a kifejlesztett szoftver alkalmazás hibáinak észlelésére [5].

Az [7] cikk egy olyan módszert mutat be, amelyben a teszt célja a rendszer számítási teljesítményének növelése az eredmények elemzése alapján. A számítási teljesítmény megállapítását mérésekkel végezzük. A rendszer tervezésénél és megvalósításánál a következő szempontokat vettük figyelembe:

- A mérés a hardver (processzor) működését teszteli, tehát független a beágyazott processzoron (számítógépen) futó szoftvertől.
- A mérést a processzor architektúra módosítása nélkül kell végrehajtani.
- A mérést valós időben kell végrehajtani, hogy automatikusan reagálhasson a szoftver viselkedésében bekövetkezett változásokra.

A kidolgozott módszer elsősorban FPGA áramkörökön megvalósított beágyazott rendszerek számítási teljesítményének elemzésére és növelésére alkalmas. Ezeknek a rendszereknek az architektúrája lehetővé teszi a fenti, az [7]-ben leírt szempontok figyelembevételét.

Jelen cikk a következő felépítést követi: a bevezetést követően bemutatásra kerül a mérési módszer, majd a Rendszermodellezés fejezetben a megtervezett modellt és a kapott eredményeket, végül a következtetéseket ismerteti.

2. MÉRÉS MÓDSZER

A processzor működése során megkülönböztetünk két állapotot: amikor a processzor futtatja a programokat – ezt nevezzük terhelt állapotnak; míg lehetségesek olyan állapotok amikor a processzornak nem kell aktívnak lennie – ezt nevezzük terheletlen állapotnak. A terheletlen állapotnál megkülönböztetjük:

- az alvó üzemmódot (sleep state), amikor a processzor nem végez számítási műveletet és úgy tűnik mintha ki lenne kapcsolva – a processzor disszipált teljesítménye nagyon kicsi;
- a tétlen állapot (idle state), amikor a processzor program végrehajtása közben várakozik valamilyen eseményre.

A processzor aktív állapotában, a szoftver aktivitástól függően a rendszer különböző részei nagyobb mértékben aktívak mint a processzor egyéb részei, például az aritmetikai egység -- amikor matematikai számításokat végez, vagy a memória illesztő egység a ki-bementi műveletek elvégzése során. Ennek alapján a következő processzor működési kategóriákat különböztetjük meg: alap (terheletlen állapot); számítás intenzív, és ki-be műveletvégzés. Ezt a fajta osztályozást a processzor különböző részeinek tényleges terhelésének mérésével végeztük. A mérések segítségével azonosítottuk az intenzívebben használt processzor részeket.

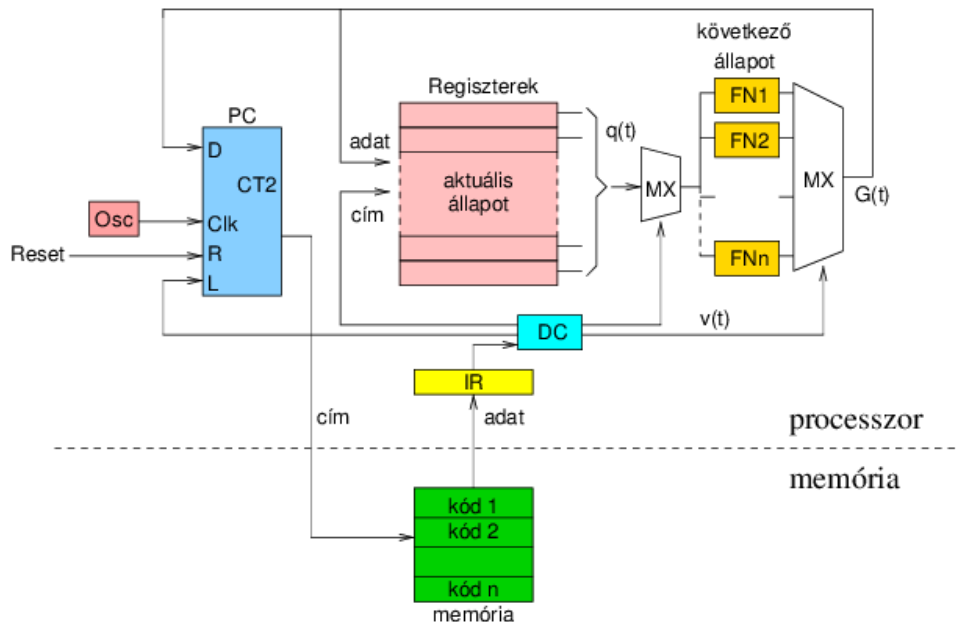
2.1. Mérési módszerek

Amint azt már említettük, a mérést a processzor szerkezetének módosítása nélkül kell elvégezni és nem befolyásolhatja a feldolgozási sebességet. A felhasznált processzor esetében az alkatrészek csatlakozási pontjai (be- és kimenetei) hozzáférhetőek, így a megjelenített jelek a mérést és osztályozást végző áramkörre csatlakoztathatók. Az FPGA áramkörökben egy lágymagos processzor paraméterezhető "Intellectual Property" (IP) modul alkalmazásával, vagy hardverleíró nyelv, vagyis az RTL (Register Transfer Level) leírás segítségével. Jelen cikkben bemutatott módszer csak RTL formátumban megadott processzorokra alkalmazható [6]. A mérési módszert az alkalmazott processzor belső felépítése segíti (1. ábra). A processzor

egy olyan állapot gép, amelyben a tárolóelemek (az ábrán: Regiszterek) biztosítják a rendszer aktuális állapotának megőrzését. Az aktuális állapot (q) az órajel ütemében változik. Az új állapot értékét az ún. gerjesztő hálózat (az ábrán FN1, FN2, stb.) állítja elő az aktuális állapot és az esetleges külső jelek (v) függvényében. A v , G , q értékei az idő függvényei (ebben az esetben az állapotot szabályozó órajel változik). Mivel a tárolt érték a gerjesztési értékévé válik egy órajel késleltetés után [7], ezért:

$$q(t) = G(t-1) \quad (1)$$

ahol az időt (t), a működtető órajel periódusainak számával mérjük.



1. ábra. Központi egység modell

A külső jelek befolyásolják a G értékeket, így nem szükséges közvetlenül tesztelni őket. A G értékek egy órajel késleltetéssel kerülnek át a q (következő állapot -- next state) értékekre, így a mérés akár a G , akár a q jelek vizsgálatával elvégezhető. A CPU egyes összetevőinek működését a hozzájuk tartozó memória értékének változása jelzi. Tehát a vizsgálat szempontjából releváns q jelek mérhető jellemzője a q jelek változási gyakorisága, vagyis a jel frekvenciája. Az osztályozás azonban nem a jel frekvenciák pillanatnyi értékein alapul, hanem a frekvencia időbeli változásán, a frekvencia időfüggvényén: $f_i(t)$.

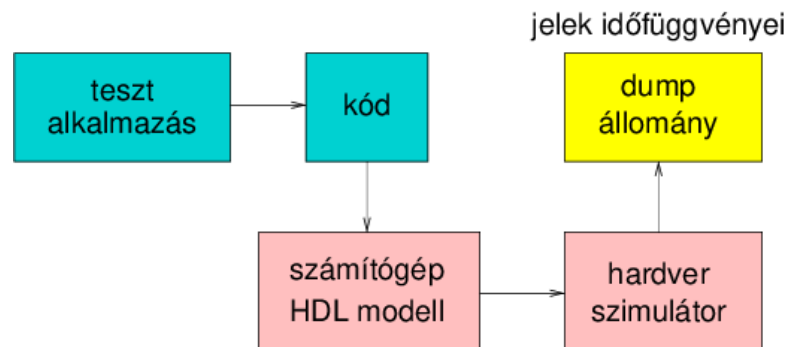
3. A TESZT RENDSZER ISMERTETÉSE

Az alkalmazás, amelyen a processzor terhelést mérését elvégeztük, egy olyan FPGA áramkörön megvalósított beágyazott rendszer, amely egy a Nap mozgását követő eszközt vezérel. Az alkalmazás egy napelem rendszer, amely a Nap mozgásának függvényében mindig az optimális működést állítja be, azaz a maximális felületet mutatja a Nap felé, amely merőleges a napsugarakra. A vezérlőszoftver az idő és a földrajzi koordináták függvényében kiszámítja a Nap sugarak pillanatnyi tájolását (vízszintes és északi szögét). A kiszámított koordináták alapján beállítja a napelem panelt, hogy merőleges legyen a napsugarakra, azaz a motorok vezérlésével mozgatja a panelt. A számítás eredményétől függő mozgatás (panelek beállása) során a szoftver nagyon sok I/O műveletet végez, ugyanez a helyzet a kezelőfelület kijelzőjének a frissítése során is.

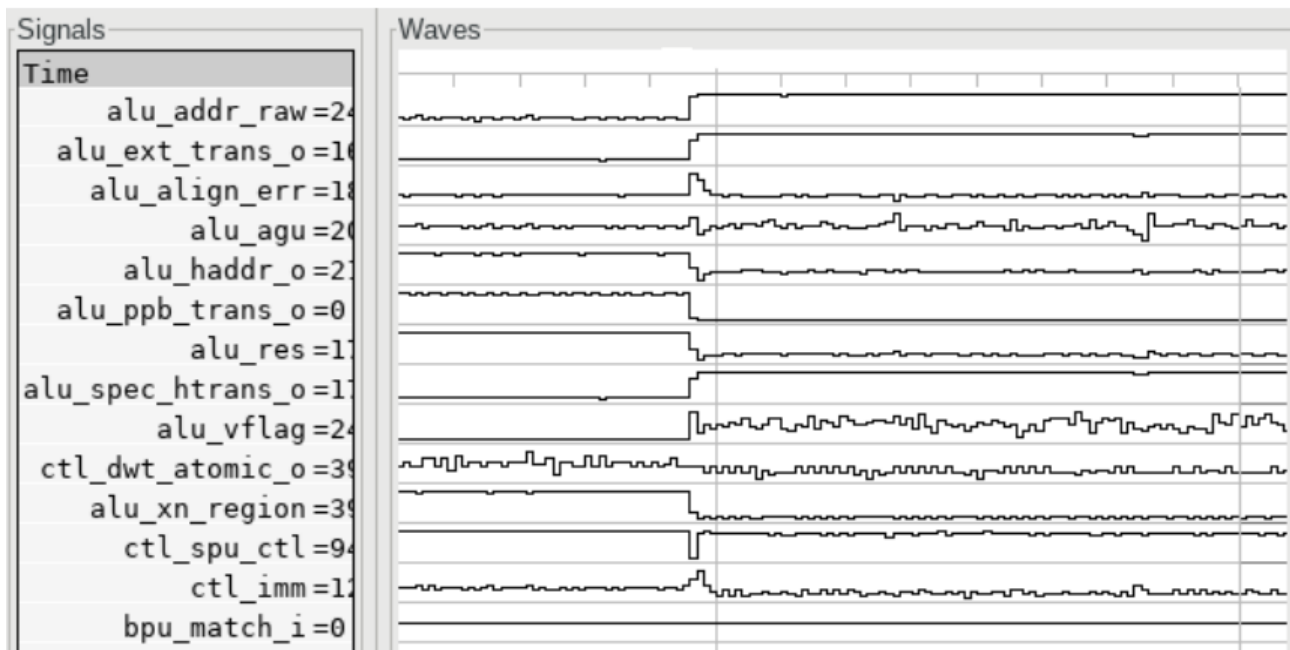
4. MÉRÉSI EREDMÉNYEK

A rendszer modellezését FPGA-n megvalósított beágyazott rendszerrel végeztük, RTL alapú ARM M0+ processzorral (2. ábra). A vizsgálat során a modellt hardver szimulátor segítségével futtatjuk, amely az áramköri elemek ki- és bemenetein megjelenő jelek értékeit lementi egy úgynevezett dump állományba, amelynek a tartalma megfelelő szoftverekkel vizsgálható.

Az $f_r(t)$ frekvenciajel időbeli változását egy dedikált szoftver állítja elő, amely feldolgozza a szimuláció kimeneti állományát. A feldolgozó szoftverbe épített algoritmus megfelel a frekvenciaméréshez használt digitális áramkör működésének. A mért frekvenciaértékeket egy VCD (Value Change Dump) formátumú fájlba írja a mérő rendszer. Ez az állomány tartalmazza az áramkör összes jelének frekvencia függvényét. A VCD fájl ábrázolásával ellenőrizhető, hogy mely frekvenciajel változások mutatnak érzékelhető eltérést valamilyen jellemzőben abban az időpontban, amikor az alkalmazás szoftver megváltoztatja a processzor számítási állapotát (terhelését), azaz amikor a számítási algoritmus elindul vagy leáll. A 3. ábra néhány mért jel változását mutatja be az alkalmazási program számítási részének elindulása pillanatában. A 3. ábrán látható jelváltozást vizsgálva például az *alu_ext_trans_o* jel esetében észrevehető, hogy jelentős amplitúdóváltozást mutat, ami azt jelenti, hogy a processzor elkezdte a számítást.



2. ábra. A rendszer modell vizsgálata



3. ábra. Mérési Eredmények [7]

A modellezés és szimulációs eredmények alapján kiválaszthatók azok a jelek, amelyek alkalmasak a központi egység számítási terhelésének mérésére. A modell esetében ismert, hogy mikor végez az alkalmazói szoftver olyan számítást, amely a CPU terhelését okozza. Az $f_r(t)$ függvények statisztikai jellemzőinek vizsgálatával megállapítható, hogy melyik mutat nagyobb mértékű korrelációt a detektálandó processzor viselkedéssel. A beágyazott rendszer struktúrájának módosítása után a kiválasztott jeleket kiveztették a processzorból. Ezen jeleket pedig a mérő rendszer segítségével elemeztük. A jelek frekvencia mérése után egy amplitúdó változást figyelő modul segítségével megállapítható a jelváltozás amplitúdója. Az amplitúdó változásokból pedig következtethető a processzor számítási terhelése.

IRODALMI HIVATKOZÁSOK

- [1] F. Vahid, T. Givargis, “Embedded system design, a unified hardware/software introduction,” Wiley and Sons, ISBN 0-471-38678-2, 2002, pp. 324.
- [2] Drótos, J. Vásárhelyi, “Interrupt driven parallel processing,” 2019 20th International Carpathian Control Conference (ICCC), Krakow-Wieliczka, Poland, doi: 10.1109/CarpathianCC.2019.8765909. 2019, pp. 1-4.
- [3] H. Sayadi, N. Patel, S. M. P.D., A. Sasan, S. Rafatirad, H. Homay-oun, “Ensemble learning for effective run-time hardware-based malware detection: A comprehensive analysis and classification,” 55th ACM/ESDA/IEEE Design Automation Conference (DAC), DOI:10.1109/DAC.2018.8465828, 2018, pp. 1-6.
- [4] Guoliang Zhao, Safwat Hassan, Ying Zou, Derek Truong, Toby Corbin, “Predicting performance anomalies in software systems at run-time,” ACM Trans. Softw. Eng. Methodol. 30, 3, Article 33 (July 2021), 33 pages. DOI:10.1145/3440757
- [5] Fletcher, Bryan H., “FPGA embedded processors,” Embedded Training Program Embedded Systems Conference San Francisco 2005 ETP-367, 2005, pp. 37.
- [6] Drótos D., “Soft processor implemented on FPGA,” <http://mzsola.iit.uni-miskolc.hu/d/softproc>, 2016
- [7] D. Drótos and J. Vásárhelyi, Microprocessor load measurement in an embedded system, 2023 24th International Carpathian Control Conference (ICCC), Miskolc-Szilvásvárad, Hungary, 2023, DOI: 10.1109/ICCC57093.2023.10178929, 2023, pp. 110-113.