Multidiszciplináris tudományok, 10. kötet. (2020) 4 sz. pp. 315-322 **https://doi.org/10.35925/j.multi.2020.4.35**

A TMS320C6000 DIGITÁLIS JELPROCESSZOROK HARDVERES ÉS SZOFTVERES FEJLESZTŐESZKÖZEI

Varga Attila Károly
egyetemi docens, Miskolci Egyetem, Automatizálási és Infokommunikációs Intézet
3515 Miskolc, Miskolc-Egyetemváros, e-mail: varga.attila@uni-miskolc.hu

Absztrakt

A digitális jelprocesszorok (DSP) gyártása a megjelenésük óta külön iparággá fejlődött. A DSP-k kifejlesztéséhez hozzájárult a digitális módszerek elterjedése a telekommunikációban és ennek következtében az olcsó programozható jelfeldolgozó eszközök iránti kereslet növekedése. A digitális jelprocesszorok fontossága és fejlődése piaci forgalmukkal rendkívül jól jellemezhető. A fejlődés rendkívül gyors volt, mely a hardver mellett a szoftverfejlesztés terén is megfigyelhető volt. A jelprocesszorok folyamatosan bővítik a digitális jelfeldolgozás alkalmazási területeit és lehetőségeit, melyek funkcionalitásából eredően a rendelkezésre álló fejlesztőeszközök is folyamatosan bővülnek. Növekvő fontosságának megfelelő súllyal kell szerepelnie. Sokféle felhasználóbarát integrált fejlesztői környezet áll a felhasználók rendelkezésére alkalmazásfejlesztésre és szimulációra egyaránt, és természetesen már grafikus fejlesztői környezetben is programozhatjuk a jelprocesszorokat. A cikk a TMS320C6000 digitális jelprocesszorok hardveres és szoftveres fejlesztőeszközeit mutatja be.

Kulcsszavak: *jelfeldolgozás, jelprocesszor, integrált fejlesztőkörnyezet, szoftverszimulátor, starter kit, emulátor*

Abstract

The manufacture of digital signal processors (DSPs) has evolved into a separate industry since their inception. The development of DSPs has been facilitated by the proliferation of digital methods in telecommunications and, as a result, the growing demand for low-cost programmable signal processing devices. The importance and development of digital signal processors can be extremely well characterized by their market turnover. Development has been extremely rapid, which can be observed not only in hardware but also in software development. In terms of signal processors, the fields of application and possibilities of digital signal processing are constantly expanding, and due to their functionality, the available development tools are constantly expanding. its growing importance should be given due weight. A wide variety of user-friendly integrated development environments are available to users for both application development and simulation, and of course signal processors can be programmed in graphical development environments. This goal of this publication is to describe the hardware and software development tools for TMS320C6000 digital signal processors.

Keywords: *signal processing, digital signal processor, integrated development environment, software simulator, starter kit, emulator*

1. Bevezetés

A TMS320C6000 digitális jelprocesszorok hardver- és szoftver-fejlesztőeszközök széles skálájával rendelkeznek. Ezek közül a C/C++, valamint az assembly fordítóprogramok, szerkesztők, könyvtárkezelők,

szoftverszimulátor, optimalizáló programok és más segédprogramok, valamint az összes alkalmazásfejlesztő eszközt egyetlen rendszerbe foglaló integrált fejlesztői környezet, a Code Composer Studio képezik az alapvető komponenseket. A Code Composer Studio általános fejlesztői környezet, amely szoftverszimulátorral és különféle hardver-fejlesztőeszközökkel használható alkalmazások prototípusainak kipróbálására, és ami a leglényegesebb, a jelprocesszoros alkalmazások megvalósítására. A következőkben a legfontosabb fejlesztőeszközöket tekintem át. [1] [2] [3]

2. Szoftverszimulátorok

A szoftverszimulátorok a processzor működését a host gépen futó programmal szimulálják. Lehetővé teszik a lefordított, összeszerkesztett, futásra kész program letöltését a program futtatását és analízisét.

A Code Composer Studio szintén lehetővé teszi szoftverszimulátor használatát, amely a TMS320C62x, C64x, C67x processzorok szimulációjára alkalmas. A fejlesztőknek a következő szolgáltatásokat nyújtja:

- a TMS320C6000 CPU teljes utasításkészletének értelmezése és végrehajtása,
- port csatlakoztatása külső periféria szimulációjához,
- kivezetések csatlakoztatása külső megszakítások szimulációja céljából,
- események analízise és nyomkövetése,
- valós idejű adatcsere (RTDX™) szimulációja,
- DSP/BIOS™ valós idejű analízise,
- CPU erőforrás-konfliktusok jelzése,
- események visszajátszása.

Különböző célra kifejlesztett szimulátorok állnak rendelkezésre. Így vannak ciklus szabatos CPU, eszköz és ciklus szabatos eszközszimulátorok. Kiválaszthatjuk az alkalmazás követelményeinek és a fejlesztési fázisnak legalkalmasabb verziót, hiszen egyes esetekben a célhardver funkcionalitása, más esetekben a pontos ciklusszám meghatározása lehet a fő követelmény. A szimulátor típusa a Code Composer Studion belül menüből választható. [4]

A ciklus szabatos CPU-szimulátort akkor választjuk, ha a fő cél a program optimalizálása. Ekkor a CPU ciklusviszonyai a fontosak és e fejlesztési fázisban nincs szükségünk a teljes eszköz szimulációjára, például egy külső eszköz elérésekor fellépő késleltetésre.

Az eszközszimulátor a kulcsperifériák funkcióit modellezi, a pontos ciklusszám nem annyira lényeges. E szimulátorokat akkor használjuk, ha az egyes erőforrások lehetőségeit vizsgáljuk. Az eszközszimulátorok gyorsabbak, mint a ciklus szabatos szimulátorok.

A ciklus szabatos eszközszimulátor a ciklusszám szempontjából is pontosan modellezi az eszközök, így a perifériák és külső memóriák funkcióit. Ezért e szimulátorok az alkalmazások ciklus szabatos vizsgálatára is alkalmasak.

Külső események és adatok szimulációja

Valós hardver környezetben a DSP külső jeleken keresztül kerül kölcsönhatásba a környezetével. A külső jelek kétfélék lehetnek:

- *Vezérlőjelek*, amelyek valamilyen aktivitást váltanak ki a jelprocesszorban. Tipikus ilyen jelek a megszakítások és periféria órajelek.
- *Adatok*, a memóriák, perifériák és a processzor között.

A szimulátor ezek modellezésére a *pin connect* és *port connect* lehetőségekkel rendelkezik.

PIN Connect (láb csatlakoztatás)

Lehetővé teszi a megszakítások szimulációját. A lábhoz speciális formátumú fájlt csatlakoztathatunk, amellyel impulzust vagy jelet állíthatunk elő. Néhány a lehetséges jel és kivezetések megadásából:

NMI	Nem maszkolható megszakítás	Impulzus
INT4	Általános célú külső megszakítás	Impulzus
INT5	Általános célú külső megszakítás	Impulzus
INT6	Általános célú külső megszakítás	Impulzus
INT7	Általános célú külső megszakítás	Impulzus
TINP0	Timer0 bemenet	Jelforma
TINP1	Timer1 bemenet	Jelforma
FSX0	McBSP0 adás frame szinkron	Jelforma
FSR0	McBSP0 vétel frame szinkron	Jelforma

A külső megszakítás szimulációjához előbb létre kell hozni az adatfájlt, amely specifikálja a megszakítást, majd a Code Composer Studio pin Connect Tools menüjéből csatlakoztatni kell az adatfájlt. Ezután betöltjük az alkalmazást és futtatjuk. Az adatfájl egy lehetséges formátumára példa az alábbi:
10 (+5 +20) rpt EOS

A megszakítások a 10., 15. (10 + 5), a. 35. (15 + 20), 40., (35 + 5), 60. (40 + 20), … gépi ciklusban lépnek fel, és a szimuláció végéig (EOS) ismétlődnek. Az időpontok itt relatív értékekkel vannak megadva, de használhatunk abszolút időértékeket is. [5] [6] [7]

Port connect

A Port Connect fájl egy vagy több sort tartalmazhat, soronként maximum 80 karakterrel. A Port Connect fájlt a szimulátor hexadecimális adatként értelmezi, amely opcionálisan kezdődhet 0x karakterekkel. Egy ilyen fájl például a következő tartalmú lehet:

```
12346666
33449999
;This is a commented line
5655cccc ; This is a commented sentence
89897f7f
```

Tiltott memória-hozzáférés detektálás

Ha az alkalmazói program különböző eszközök által foglalt memóriaterületekhez fér hozzá, a program viselkedése kiszámíthatatlan lehet. A szimulátor hibaüzenet küldésével jelzi a tárterület megsértését. Ha a szimulátor perifériaeszközt is modellez, a szimulátor azt is foglalt memóriaterületnek tekinti. A memóriaterület megsértésekor keletkező hibaüzenetre példa lehet az alábbi hibajelzés:

```
Memory Map Error: WRITE access to address 0xc7fffc,
          which is RESERVED in Hardware.
```

CPU erőforrás-konfliktus detektálás

A C6000 CPU legfontosabb erőforrásait a memóriák, az A és B oldali funkcionális egységek, az adatutak és a CPU-regiszterek képezik. Ha az erőforrások használatával kapcsolatos korlátozásokat a

végrehajtott program megsérti, a program eredménye határozatlanná válik. Mivel a korlátozások megsértése csak a program futása alatt derül ki, a szimulátor hibaüzenetet ad. Ilyen üzenetsor például a következő:

- `Memory Map Error: WRITE access to address 0xc7fffc, which is RESERVED in Hardware.`
- `Memory Map Error: READ access to address 0x1880000, which is NOT SUPPORTED in Simulator.`

Szimulátoranalízis

A TMS320C6000 Szimulátor Analízis lehetővé teszi specifikus események rögzítését és kijelzését. Ilyenek lehetnek: nincs programtalálat a cache memóriában (program cache miss), programtalálat a cache memóriában (program cache hit), utasítás lehívás (program fetch), 0. és 1. blokkok elérése. Az események monitorozása alapján a program teljesítménye mérhető. Az események egy számlálót számláltathatnak, vagy leállást eredményezhetnek. A leállítási képesség teszi lehetővé az alkalmazás működésének nyomon követését. Például az időzítő szinkron jele indítja az EDMA egységet, a leállítás segítségével ellenőrizhetjük, helyes volt-e a működés. Az események számlálása egy utasításperiódus alatt felhasználható a program optimalizálásához. A szimulátoranalízis a Code Composer Studio Tools menüpontjából érhető el.

Valós idejű adatcsere (RTDX)

Ahhoz, hogy egy RTDX alkalmazást futtathassunk szimulátor alatt, az alkalmazást össze kell szerkeszteni az RTSX Simulator Target könyvtárral. Az RTDX alkalmazás egyszerűen átvihető a valós hardverre.

DSP BIOS

Minden alkalmazás, amely DSP/BIOS elemeket használ, futtatható szimulátoron is. Ha azonban az alkalmazás valós idejű analízise szükséges, a projekt konfigurációban RTDX módot kell beállítani. [8]

Bootload

Bootloadnak nevezzük azt a folyamatot, amikor a boot módban meghatározott címről véges számú szavakat másolunk a 0x0 címre. Szimulátor esetén bootload akkor történik, ha a szimulátor konfigurációs fájljában boot módot engedélyezünk.

Felhasználóimemória-használat detektálás

Lehetőség van arra is, hogy a program által felhasznált memória nagyságát ellenőrizzük. A szimulátor hibajelzést küld, ha az alkalmazás által használt memória nagysága meghalad egy megadott limitet. Alapértelmezés szerint ez 64 MB. Ez a konfigurációs fájlban megadható nagyobbra is.

EMIF órajel

A Code Composer Studio setup segítségével az EMIF és a CPU órajel frekvencia programozható. A frekvenciát az adott DSP fejlesztőkártyának megfelelően (DSK/TEB/EVM) kell beállítani. A beállítás nem minden szimulátornál lehetséges, de a C6711/C6712/C6713 ciklus szabatos eszköz szimulátorok esetén be kell állítani.

Események visszajátszása (Rewind)

A funkció a C67xx ciklus szabatos CPU Szimulátor és a C6713 eszközszimulátor esetén lehetséges.

Segítségével a végrehajtott alkalmazás futásának története újra megtekinthető. Ez lerövidítheti a nyomkövetés idejét.

A szimulátor konfigurálása alatt az alábbi paraméterek beállatását értjük:

- erőforráskonfliktus-detektálás mód,
- foglalt memóriaterület megsértése detektálás mód,
- bootload mód,
- EMIF és CPU órajel frekvenciák,
- rewind engedélyezése,
- McBSP XBAR,
- McASP XBAR,
- használható memóriaterület limit,
- fájl formátum Pin Connect szimulációhoz,
- fáj1 Formátum Port Connect szimulációhoz,
- alap konfigurációs fájl.

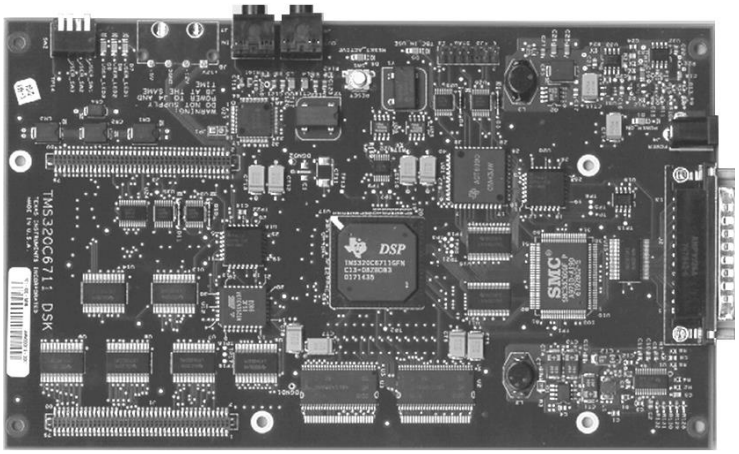
A Code Composer Studio installálási alkönyvtárában találhatók a szimulátorok konfigurációs fájljai, amelyek alapértelmezés szerinti konfigurációs paramétereket tartalmaznak. E paraméterek a Code Composer Studio Setup programból módosíthatók. [7]

A C6713 eszközszimulátor konfigurációs fájlt az alábbiakban mutatjuk be. A fő paraméterek a következők:

- dinamikus könyvtár: spkc6x1x.dll,
- jelprocesszor típusa: C6713,
- boot mód tiltás,
- az EMIF órajel frekvenciája a CPU órajelek frekvenciájának 0,6-szerese
- erőforrás konfliktus detektálás,
- memóriatérkép-ellenőrzés,
- használható memórialimit: 64 MB.

3. Starter Kitek

A DSP Starter Kit prototípusok, alkalmazások demo verziói fejlesztésére, koncepciók ellenőrzésére és nem utolsósorban oktatási célokra közkedvelt eszköz. A legtöbb Starter Kit DSK verziójú Code Composer Studio fejlesztőrendszert is tartalmaz. A TMS320C6711 és C6713 jelprocesszoros digitális starter kitek hardver kártyája a digitális jelprocesszoron kívül JTAG emulátor vezérlőt, JTAG csatlakozót, USB vagy párhuzamos port interfészt, 16 MB SDRAM és 128 KB FLASH memóriát, 16 bites CODEC analóg interfészt, analóg bemenetet és kimenetet, rendszerbővítő csatlakozót, a felhasználói program által elérhető LED diódákat és kapcsolókat, valamint hálózati tápegységet és USB vagy párhuzamos interfész kábelt is tartalmaz. A rendszerbővítő csatlakozón keresztül kiegészítő kártyák (Daughter Card) kapcsolhatók a starter kithez, ami sokféle alkalmazás tesztelését teszi lehetővé.



1. ábra. *A TMS320C6711 DSK*

4. Alkalmazásfejlesztő modulok

Az alkalmazásfejlesztő, vagy kiértékelő modul, (Evaluation Module, EVM) olyan hardver fejlesztőeszköz, amely digitális jelprocesszort, perifériákat, különböző kiépítésű memóriákat, bővítő csatlakozókat és JTAG interfészt tartalmaz. Lehetővé teszik az alkalmazói szoftverek, analízisét, a programfutás kiértékelését, nyomkövetését és a prototípus elkészítését. Mielőtt a fejlesztő elkezdené a saját hardver rendszer megtervezését, megbizonyosodhat a tervezési koncepciója helyességéről, ami lerövidítheti a fejlesztési időt és csökkentheti a fejlesztési költségeket. Az alkalmazásfejlesztő modulok a starter kitekhez képest több erőforrással rendelkeznek és általában valamilyen alkalmazásnak megfelelő konfigurációval rendelkeznek, ezért tulajdonképpen bizonyos alkalmazások szempontjai szerint előre definiált hardver rendszereknek tekinthetők. A host számítógéphez vagy emulátoron keresztül, vagy közvetlenül a PC buszrendszerén át csatlakoznak. Az EVM modulok ára tág határok között változik, de mindig nagyobb a Starter Kitek áránál, egy nagyságrend különbség is lehet. A különbségnek megfelelően az EVM modulokhoz biztosított fejlesztői környezet is teljesebb szolgáltatást nyújt. [2] [3]

Az EVM szoftver rendszere két részből áll: host PC szoftver és DSP szoftver. A host szoftver komponensei: kártya konfigurációs segédprogramok, Code Composer fejlesztőrendszer, EVM tesztprogramok, C6x COFF betöltő segédprogramok, Win32 host könyvtárak, példaprogramok. A DSP szoftver elemei: perifériameghajtó programok, célhardver specifikus könyvtár, forrásnyelvű példaprogramok az erőforrások használatához.

5. Emulátorok

Az áramkör emulátorok (In-Circuit emulator, ICE, on-circuit debugger, OCD, vagy background debug monitor, BDM) hardver eszközök, amelyeket a beágyazott rendszerek szoftvereinek nyomkövetésére használnak. A beágyazott rendszerek fejlesztése ugyanis speciális problémát jelent, mivel nem rendelkeznek felhasználói interfésszel, monitorral, klaviatúrával, diszk egységekkel, ami az általános számítógépek esetében természetes. A legtöbb áramkör-emulátor teljes órajel frekvenciával működik. Ezenkívül teljes mértékben transzparens, ami lehetővé teszi, hogy mi vezéreljük a program működését: elindíthatjuk és leállíthatjuk, lépésenként futtathatjuk a programot, töréspontokat helyezhetünk el a programban, kijelezhetjük és módosíthatjuk a regiszterek és a memória tartalmát.

Az in-circuit emulációt hardver emulációnak is nevezik abban az esetben, ha a fejlesztés alatt lévő rendszerben az emulátort a processzor helyére csatlakoztatják. Az első jelprocesszorok megjelenése idején az emulátorok ilyen módon működtek. Bonyolult és drága áramkörök voltak, amelyek a fejlesztés ideje alatt betöltötték az ideiglenesen eltávolított processzor feladatát. [4] [5] [6] [9]

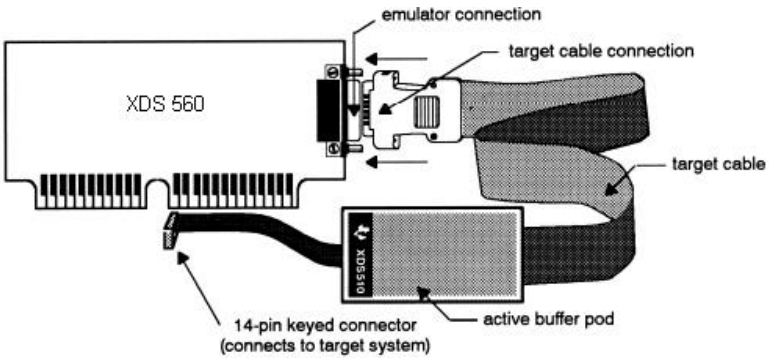
JTAG emulátorok

A JTAG emulátor szabványos, egyszerű csatlakozást biztosít a fejlesztőrendszer és egy vagy több célhardver között. Attól kezdve, hogy a DSP kártyára a tápfeszültséget rákapcsoltuk, a CPU és az összes erőforrás állapota nyomon követhető.

Az XDS 560 emulátor PCI kártyából és a célhardvert csatlakoztató felhasználói kábelből áll. A hardver nyomkövetése a processzor leállításával történik, ami lehetővé teszi az információk lekérdezését a CPU-ból, illetve információk beírását a CPU-ba a JTAG csatlakozón keresztül. Az információ átvitele sorosan történik, az IEEE 1149.1 specifikációnak megfelelően. Ez a nyomkövetési módszer közel valós idejű, de behatoló típusú (intrusive), mivel a CPU leállításával jár. Bár a JTAG csatlakozó azonos a peremfeltételes tesztcsatlakozóval, az emuláció folyamata a teszteléstől eltérő. A processzoron belül különféle adatletapogatási láncok vannak, és az emulátorkártya vezérli, melyik adatletapogatási lánc működik és milyen információkat tartalmaznak a láncok. A Scan menedzsernek nevezett funkció vezérli az adatok letapogatását vagy beírását. Ezenkívül gondoskodik az adatátvitelről a processzor és a nyomkövető program képernyője között.

PC-s fejlesztő rendszer esetén a célprocesszor JTAG csatlakoztatásra több lehetséges módszer is van. A célhardver csatlakozhat a Host géphez PCI, ISA kártyán, vagy USB, PCMCIA, illetve Etherneten keresztül. A különféle csatlakoztatási módot dinamikus könyvtárak támogatják.

Az XDS 560 emulátor PCI kártyából és a célhardvert csatlakoztató felhasználói kábelből áll, amint a 2. ábra szemlélteti.



2. ábra. *XDS560 emulátor hardver elemei*

Az XDS 510USB emulátor a PC USB portjára csatlakozik. Az alábbi Texas Instruments processzorok fejlesztésére alkalmas: F240x, F28xx, C54xx, C55xx, C6xxx, TMS470 ARM9, OMAP, DaVinci™. Tápfeszültsége az USB portról biztosítható, nem igényel külön tápegységet. Kompatibilis a Code Composer Studio legújabb verzióival. Windows 98 SE, 2000, XP operációs rendszerek alatt használható. A fejlesztő szoftver Flash Programozó segédprogramokat is tartalmaz. A 14 érintkezős JTAG csatlakozóaljzatot a célhardveren a processzorhoz a lehető legközelebb kell elhelyezni. A JTAG csatlakozó jelei:

- TRST: Teszt mód Reset,
- TMS: Teszt mód kiválasztás,

– TDI: Teszt adat bemenet,
– TDO: Teszt adat kimenet,
– TCK: Teszt Clock .

6. Összefoglalás

Mint minden jelprocesszornál, a TMSC6000 DSP-nél is fontos szempont a rendelkezésre álló fejlesztői környezet fejlettsége, melyhez számos integrált fejlesztőkörnyezet áll rendelkezésre a fejlesztésre és szimulációra egyaránt, és természetesen már grafikus fejlesztői környezetben is programozhatjuk a jelprocesszorokat. A korszerű szimulátorok a hardver komponenseket is tartalmazó teljes fejlesztőrendszerek szolgáltatásainak nagy részét biztosítják. A DSK kedvező árú hardverfejlesztő eszköz, melynek működése könnyen elsajátítható, és alkalmas a digitális jelprocesszor teljesítőképességének tesztelésére. Az alkalmazásfejlesztő modul lehetővé teszi az alkalmazói szoftverek, analízisét, a programfutás kiértékelését, nyomkövetését és a prototípus elkészítését, az emulátorok pedig lehetővé teszik, hogy az alkalmazás futását valós körülmények között teszteljük. Bár a jelprocesszorok az utóbbi néhány évtizedben páratlan fejlődésen mentek keresztül, néhány alapvető jellemzőt a jelenlegi legkorszerűbb processzorok napjainkig megőriztek. A C6000 architektúrának megfelelően a memóriaírás és -olvasás a memória és a regiszterek közötti adatmozgatást jelenti. Az assembly kódnak megfelelően minden utasítás cél operandust és egy vagy két forrás operandust tartalmaz. A TMS320C6000 C nyelv az ANSI szabványnak felel meg, amely megenged néhány eltérést a szabványos C rendszertől. A TMS320C6000 DSP gyakorlatilag a host és a célhardver sajátosságait veszi figyelembe, melynek hatékonyságnövelési és gyakorlati okai vannak.

7. Köszönetnyilvánítás

A cikkben ismertetett kutatómunka az EFOP-3.6.1-16-2016-00011 jelű *Fiatalodó és Megújuló Egyetem – Innovatív Tudásváros – a Miskolci Egyetem intelligens szakosodást szolgáló intézményi fejlesztése* projekt részeként – a Széchenyi 2020 keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Irodalom

[1] Dahnoun, N.: *Digital Signalprocessing implementation using the TMS320C6000™ DSP Platform*, Pentice Hall 2000, ISBN 0201-61916-4.
[2] Chassing, R.: *DSP Applications using C and the TMS320C6x DSK*, John Wiley Sons, Inc.
[3] TMS320C6000 Assembly Language Tools User's Guide; Literature Number: SPRU186G, January 2000.
[4] TMS320C6000 Optimizing Compiler User's Guide; Literature Number: SPRU187 March 2000.
[5] TMS320C6000 CPU and Instruction Set Reference Guide; Literature Number: SPRU189E January 2000.
[6] TMS320C6000 Programmer's Guide; Literature Number: SPRU198D March 2000.
[7] TMS320C6000 Code Composer Studio Tutorial; Literature Number: SPRU301C February 2000.
[8] TMS320C6000 DSP/BIOS User's Guide; Literature Number: SPRU303B March 2000.
[9] Rongen, H.: *Introduction to Digital Signal Processors (DSPs)*, Forschungszentrum Jülich Zentrallabor für Elektronik 52425 Jülich, Germany.