김일성종합대학학보
(자연과학)
주체104(2015)년 제61권 제12호

JOURNAL OF **KIM IL SUNG** UNIVERSITY
(NATURAL SCIENCE)
Vol. 61 No. 12 JUCHE104(2015).

TMS320C6x계렬 DSP응용프로그람적재의 한가지 방법

왕상덕, 차영숙

매몰형통신체계에서 DSP는 핵심적인 역할을 하며 따라서 DSP응용프로그람을 작성하고 실행시키는 문제는 체계개발에서 선차적인 의의를 가진다.[1－3]

론문에서는 DSP개발도구가 없이 TMS320C6x DSP의 응용프로그람을 내리적재하고 시동시키는 방법에 대하여 서술한다.

1. DSP체계개발과정

DSP체계개발은 표준적으로 다음과 같은 단계로 이루어진다.

① DSP처리기의 선택

② DSP기술 및 도구의 평가

③ 초기개발

④ 제품개발

초기개발과 제품개발단계에서는 다같이 개발도구(Kit)와 모방기가 필요한데 일반적으로 이와 같은 도구들은 가격이 비싸다.

한편 CCS(Code Composer Studio)는 TI DSP에 대한 쏘프트웨어통합개발도구로서 각이한 DSP개발도구와 모방기에 대한 구동프로그람을 장악하고 체계개발을 종합적으로 지원하고있다. 또한 DSP개발도구가 없을 때 프로그람을 작성하고 성능을 분석하기 위한 모의기도 제공하고있다.[1, 4]

만일 CCS에서 모의기를 리용하여 DSP응용프로그람을 개발한다면 CCS콤파일출력화일(xxx.out)은 DSP의 기억기2진코드배치를 그대로 반영하지 않으며 구조가 복잡하고 은폐된 형태를 가진다. 그러므로 CCS콤파일출력화일로부터 실지 실행프로그람코드부분을 추출하여 DSP기억기2진코드를 작성하여야 한다.

2. DSP/BIOS응용프로그람의 토막구성과 탐색

CCS번역기가 생성하는 코드 및 자료블로크는 재배치가능한 블로크이며 이것은 개별적인 체계들의 다양한 기억구성에 효과적으로 대응할수 있는 방법으로 된다. 이러한 블로크들을 토막(section)이라고 부른다. 번역기는 기본적으로 두가지 종류의 토막 즉 초기화토막과 비초기화토막을 창조한다. 기본 코드토막의 구성은 표와 같다.

표. 코드토막의 구성

토막이름		내　용
초기화	.cinit	초기화된 대역 및 정적변수표
	.const	초기화된 대역 및 정적상수들
	.pinit	초기시동시에 호출되는 구조체표
	.switch	switch명령을 위한 표
	.text	실행코드 및 상수
비초기화	.bss	대역 및 정적변수들
	.stack	탄 창
	.system	Malloc함수를 위한 기억기

프로그람이 련결될 때 매개 토막에는 기억배치주소가 규정되며 CCS는 DSP/BIOS응용프로그람의 토막구조를 xxx.map형태로 제공하는데 이 화일에는 토막이름과 시작주소, 길이, 초기화상태 등이 포함되여있다.

xxx.map화일의 한가지 실례를 아래에 보여준다.

```
******************************************************************************
                  TMS320C6x  COFF  Linker  PC  v5.1.0
******************************************************************************
>> Linked Fri Apr 26 18:35:34 2013

OUTPUT FILE NAME:    <./Debug/bs_bbu.out>
ENTRY POINT SYMBOL: "_c_int00"  address: 00007d00

MEMORY CONFIGURATION

         name            origin    length      used     attr    fill
----------------------  --------  ---------  --------  ----  --------
IRAM                    00000000   00030000   00008816  RWIX
CACHE_L2                00030000   00010000   00000000  RWIX
SDRAM                   80000000   00800000   00008000  RWIX
FLASH                   90000000   00040000   00000000  RWIX
FPGA                    a0000000   00004000   00000000  RWIX

SECTION ALLOCATION MAP

 output                                                  attributes/
section   page    origin      length       input sections
--------  ----  ----------  ----------   ----------------
.pinit      0   00000000    00000000     UNINITIALIZED
```

```
.switch     0     00000000     00000000     UNINITIALIZED

.hwi_vec    0     00000000     00000200
                  00000000     00000200     bs_bbucfg.obj (.hwi_vec)

.bios       0     00000200     00003860
                  00000200     000006e0     biosi.a62 : swi.o62 (.bios)
                  000008e0     00000440                 : hwi_disp_asm.o6 (.bios)
                  00000d20     00000320                 : prd.o62 (.bios)
                  00001040     000002a0                 : sem_pend_inst.o (.bios)
                  00002900     00000100                 : gbl_cslcacheini (.bios)
                  00003800     00000060                 : sem_pend_asm_in (.bios)
                  00003a00     00000020                 : sts_set.o62 (.bios)
                  00003a20     00000020                 : fxn_c.o62 (.bios)
                  00003a40     00000020                 : utl_halt_inst.o (.bios)
…
```

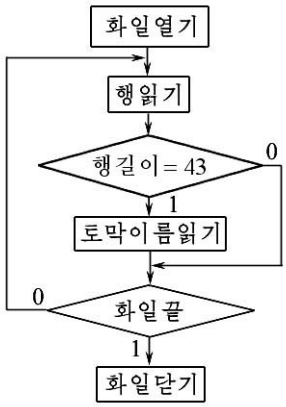

그림 1. 코드토막탐색 알고리듬

xxx.map화일에서 초기화토막에는 'UNINITIALIZED'표식이 없으므로 토막정보를 나타내는 행의 길이는 43이다. 여기에 기초한 탐색알고리듬은 그림 1과 같다.

이와 같이 xxx.map에서 초기화토막을 탐색함으로써 어느 토막이 DSP기억기에 적재되여야 하는가를 알수 있으며 여기로부터 xxx.out에서 이 토막부분을 추출하여 DSP에 적재시킨다.

이러한 xxx.out는 크게 코드토막기록과 코드부분으로 구성된다고 볼수 있는데 하나의 코드토막기록은 0x30(48)크기로 되여있다.(그림 2)

Null	토막이름	토막주소	토막주소	토막크기	토막편이주소	Null

그림 2. xxx.out의 코드토막기록형식

그림 2에서 보는바와 같이 토막주소마당은 2번 반복되여있는데 토막편이주소가 바로 주어진 토막에 대응한 화일편이주소이다. 그러므로 xxx.map에서 탐색한 코드토막에 대하여 토막편이주소를 구하고 그 위치의 코드토막을 DSP기억기의 주어진 주소에 적재하여야 한다.

3. DSP응용프로그람적재 및 시동

DSP응용프로그람은 HPI(Host Port Interface)를 조종하는 호스트(MCU)에 의하여 내리적재된다. 이때 코드토막들은 1개의 MCU패키트크기를 초과하므로 HPI_WR패키트와 HPIWR_RUN패키트로 분할하여 전송하여야 한다. 1개 토막에 대한 적재알고리듬은 그림 3과 같다.

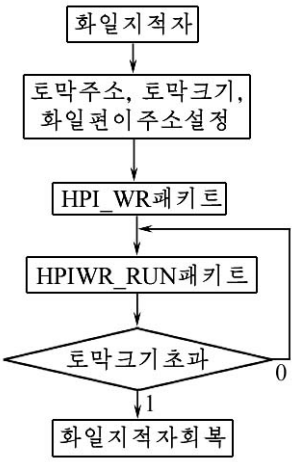

그림 3. 코드토막적재 알고리듬

코드토막적재는 xxx.out화일에서 토막기록마당을 탐색하는 도중에 초기화토막을 찾았을 때 진행되므로 먼저 현재의 화일지적자를 보관하고 토막의 화일편이주소로 넘어가야 한다.

MCU패키트의 HPI_WR는 주소마당을 가지고있으므로 코드토막기록의 주소마당값을 리용한다. HPI_WR패키트가 전송된 다음에는 코드토막크기에 이를 때까지 HPIWR_RUN패키트가 전송된다. 이때 코드토막이 다 전송되면 화일접근점은 다시 회복되여야 한다.

적재된 DSP응용프로그람은 ROM방식과 HPI방식으로 시동된다.

HPI시동방식에서 CPU는 정지상태에 들어가며 호스트는 HPI를 통하여 임의의 기억령역과 각종 조종등록기에 접근하여 필요한 초기화를 진행할수 있다. MCU는 초기화를 끝내고 0주소에 프로그람을 배치한 다음 DSPINT를 설정하여 시동공정을 완료한다. DSPINT에 의하여 CPU는 정지상태서 벗어나 0주소로부터 프로그람의 실행을 시작한다.

맺 는 말

DSP/BIOS응용프로그람에 대한 CCS콤파일출력화일(xxx.out)은 DSP의 기억기2진코드배치를 그대로 반영하지 않으며 구조가 복잡하고 은폐된 형태를 가지고있다. 그러므로 CCS콤파일출력화일로부터 실지 실행프로그람코드부분을 추출하여 DSP기억기2진코드를 작성하여야 한다. 따라서 CCS콤파일출력화일의 구조를 분석하고 코드토막정보에 의하여 TMS320C6713의 응용프로그람의 내리적재와 그 시동방법에 대하여 서술하였다.

참 고 문 헌

[1] DSP Selection Guide, Texas Instruments, 46～65, 2007.

[2] TMS320C6000 Peripheral Reference Guide, Texas Instrumnents, 342～389, 2004.

[3] TMS320 DSP/BIOS User's Guide, Texas Instruments, 213～245, 2007.

[4] Rulph Chassaing; Digital Signal Processing and Application with TMS320C6713 DSK, John Wiley & Sons, 64～97, 2008.

주체104(2015)년 8월 5일 원고접수

A Method for Downloading and Booting of TMS320C6x DSP Application

Wang Sang Dok, Cha Yong Suk

CCS compiler's output file(xxx.out) for DSP/BIOS application does not have structure of binary image of DSP memory and its structure is complex and scrambled type. Therefore, the DSP memory binary image of application should make by extracting the part of application code from CCS compiler output.

In this paper, we analyzed the structure of CCS compiler's output file and described a method for downloading and starting TMS320C6713 application using code section informations.

Key words: DSP, download, boot