DSP Instruction Cache Performance on the OMAP5912

Elvis Zapalac, Harry Thompson *SDS Technical Support*

ABSTRACT

This project benchmarked the DSP/BIOS Link loop example, which is available for the OMAP5912 OSK. It determined the performance effect of turning on the DSP-side instruction cache (I-Cache) using the Chip Support Library (CSL). The impacts were measured using the DSP/BIOS CPU Load Graph.

Contents

1 Introduction

This report provides CPU Load Graph data on the DSP/BIOS Link loop program example using different memory configurations. It demonstrates how the placement of different sections in memory can affect the execution speed of a program as well as the effect of turning on the instruction cache. Because internal memory is limited, the OMAP5912's 24 KB instruction cache (I-Cache) can be used to achieve similar performance using external memory.

The DSP/BIOS Link loop program was written for the OMAP5912 OSK and transfers data buffers between the ARM and the DSP using the SIO_issue and SIO_reclaim DSP/BIOS APIs.

2 Test Setup and Measurements

The test setup used Code Composer Studio version 2.3 with Chip Support Library (CSL) version 3.00.05.00.

To perform the test, the original loop example project was modified to include the icache.c file, which contains the icache_Enable function. This function turns on the I-Cache using CSL. The icache_Enable function call was added to the main() function in main.c of the loop example. This enables the I-Cache at the beginning of code execution. The code for the icache_Enable function is in Appendix A. It is a C function that calls the appropriate CSL 3.x CSL_icache APIs to initialize and set up the 2-way set-associative instruction cache.

Table 1 shows the loop example variations that were tested. All were debug builds.

Table 1. Test Configurations

Test Version	Sections in External Memory	I-Cache
loop.out	-- none --	Off
loop_bios_no_cache.out	.bios	Off
loop_bios_cache.out	.bios	On
loop_text_no_cache.out	.text	Off
loop_text_cache.out	.text	On
loop_bios_text_no_cache.out	.bios & .text	Off
loop_bios_text_cache.out	.bios & .text	On

The dsplink module was loaded using dsplink.bash and the above DSP COFF files were loaded using the loopgpp ARM program.

A spin loop was placed in the DSP/BIOS user init function so that CCStudio could be launched at the beginning of code execution on the DSP. The following two lines of code were added to the project's Tconf file to enable a user init function called gblUserInit to be called at startup:

```
prog.module("GBL").CALLUSERINITFXN = 1;
prog.module("GBL").USERINITFXN = prog.extern("gblUserInit");
```

The gblUserInit function was added to the project's main.c file. The following is the spin loop function:

```
void gblUserInit()
{
    volatile int i=1;
    while(i);
}
```

Code Composer Studio Setup was set to use the GEL file for DSP/BIOS Link applications, which is called omap1610_5912_dsp_gel.gel.

After CCStudio was launched, the code execution was halted inside the spin loop. Next, the CPU Load Graph (a DSP/BIOS Real-Time Analysis tool) was opened. The CPU Load Graph displays the target CPU processing load. Figure 1 shows an example CPU Load Graph:

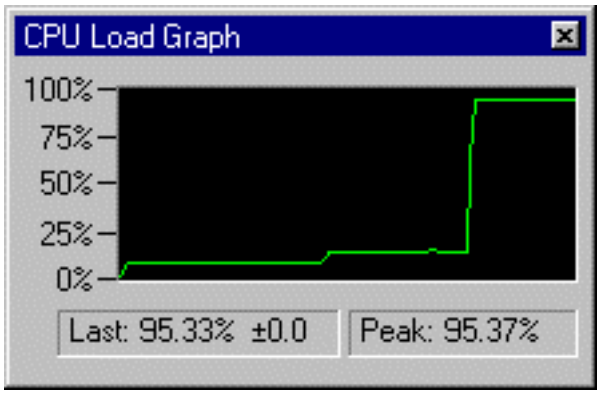

Figure 1. Example CPU Load Graph

After the CPU Load Graph was opened, the cursor was placed on the line after the while loop in the gblUserInit function. Next the Debug tab was selected and "Set PC to Cursor" was executed to get out of the while loop. Finally the code was executed and the "last" values were obtained and recorded from the CPU Load Graph. Table 2 shows the measurements provided by the CPU Load Graph.

Table 2. Test Results

Test Executed	Sections in External Memory	I-Cache	CPU Load Graph Last Values for Debug Build
loop.out		Off	26.77%
loop_bios_no_cache.out	.bios	Off	83.04%
loop_bios_cache.out	.bios	On	24.10%
loop_text_no_cache.out	.text	Off	64.09%
loop_text_cache.out	.text	On	27.04%
loop_bios_text_no_cache.out	.bios & .text	Off	96.84%
loop_bios_text_cache.out	.bios & .text	On	25.91%

3 Conclusions

1. Having the instruction cache off and code sections like .bios or .text in external memory can result in as much as a 4x performance hit. The amount of performance impact is dependent upon such factors as the task switching rate of your application, the size of the code compared to the size of the instruction cache, and the EMIF settings. For example, a telephony 8 kHz sample rate application will be less affected than a 44.1 kHz MP3 application.
2. The effects of enabling the I-Cache on the loop project are significant. Enabling the I-Cache on the loop application approximates everything being stored internally even when both the

.bios and .text sections are in external memory. Larger applications may not achieve this level of performance if the sections that are being cached are bigger than the 24 KB instruction cache.

3. By enabling the I-Cache, an additional performance improvement can be achieved because the CPU now has an additional bus to fetch instructions. For example, suppose the CPU executes from DARAM and the CPU also performs two data accesses per cycle from DARAM. If the I-Cache is enabled, the CPU could then execute from the instruction cache and perform two data accesses per cycle from DARAM. This would remove the instruction access load from the DARAM block, hence improving the performance.

4. There was little or no impact on performance between the programs that used both DARAM and SARAM and the programs that used only SARAM in the loop project. Hence the measurements for these minor differences are not listed.

4 References

1. *User Guide*, DSP/BIOS Link OSK5912 Starter Kit (OSK) MontaVista Linux Professional Edition 3.1
2. DSP/BIOS Link Download page https://www-a.ti.com/downloads/sds_support/targetcontent/link/index.html
3. OMAP 5912 Starter Kit page http://focus.ti.com/docs/toolsw/folders/print/tmdxosk5912.html
4. OMAP 5912 processor information http://focus.ti.com/omap/docs/omapgenpage.tsp?navigationId=12341&templateId=5663&path=templatedata/cm/omapproc/data/omap5912

Appendix A. icache_Enable Function

The code for the icache_Enable function in the icache.c file is as follows:

```
/*************************************************************************/
/* Function to enable I-Cache, 2-way configuration                       */
/*                                                                       */
/*************************************************************************/
#include <soc.h>
#include <log.h>
#include <icache.h>

extern LOG_Obj trace;
/* Object structure for DSP I-Cache CSL */
CSL_IcacheObj   icacheObj;

/* enable the icache for 2way */
void icache_Enable(void)
{
    CSL_Status                      status = CSL_SOK;
    CSL_IcacheHandle                hIcache;
    CSL_IcacheHwSetup               hwSetup;

    /*Init and open the DSP I-Cache Instance without hardware setup */
    hIcache = icacheInitCSL ( &status);
    if (status != CSL_SOK)
    {
        LOG_printf(&trace,"1. Error opening the instance\n");
        return;
    }

    hwSetup.disableClk = CSL_ICACHE_CFG_PARAM_ENABLE;
    hwSetup.autoGating = CSL_ICACHE_CFG_PARAM_DISABLE;

    /* Two-way set-associative */
    hwSetup.icacheOrgn = CSL_ICACHE_2WAY_ORG;

    /* Call hardware setup function to setup  */
    status = CSL_icacheHwSetup (hIcache, &hwSetup);

    if (status != CSL_SOK)
    {
        LOG_printf(&trace,"2. Cache configuration failed\n");
        CSL_icacheClose (hIcache);
        return;
    }

    /* Enable the I-Cache, write a one to CAEN bit of ST3 */
    status = CSL_icacheHwControl (hIcache , CSL_ICACHE_CMD_CACHE_ENABLE, NULL);

    if (status != CSL_SOK)
    {
        LOG_printf(&trace,"3. Cache Enable command failed\n");
        CSL_icacheClose (hIcache);
        return;
    }

  return;
}
```

```
CSL_IcacheHandle icacheInitCSL (CSL_Status* status)
{
    CSL_IcacheHandle     hIcacheLcl = NULL;

    /* Initialize the DSP I-Cache CSL */
    CSL_icacheInit (NULL);

    /* Open the DSP I-Cache Instance */
    hIcacheLcl = CSL_icacheOpen (&icacheObj, CSL_ICACHE, NULL, status);

   return (hIcacheLcl);
}
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265