



DSP/BIOS™ BRIDGE INTEGRATION DOCUMENT

Making **Wireless**

OMAP™ is a Trademark of Texas Instruments Incorporated

OMAP-Vox™ is a Trademark of Texas Instruments Incorporated

Innovator™ is a Trademark of Texas Instruments Incorporated

Code Composer Studio™ is a Trademark of Texas Instruments Incorporated

DSP/BIOS™ is a Trademark of Texas Instruments Incorporated

eXpressDSP™ is a Trademark of Texas Instruments Incorporated

TMS320™ is a Trademark of Texas Instruments Incorporated

TMS320C28x™ is a Trademark of Texas Instruments Incorporated

TMS320C6000™ is a Trademark of Texas Instruments Incorporated

TMS320C5000™ is a Trademark of Texas Instruments Incorporated

TMS320C2000™ is a Trademark of Texas Instruments Incorporated

OpenGL® is a Registered Trademark of the Khronos Group

OpenML® is a Registered Trademark of the Khronos Group

OpenVG™ is a Trademark of the Khronos Group

OpenMAX™ is a Trademark of the Khronos Group

All other trademarks are the property of the respective owner.

Copyright © 2008 Texas Instruments Incorporated. All rights reserved.

Information in this document is subject to change without notice. Texas Instruments may have pending patent applications, trademarks, copyrights, or other intellectual property rights covering matter in this document. The furnishing of this document is given for usage with Texas Instruments products only and does not give you any license to the intellectual property that might be contained within this document. Texas Instruments makes no implied or expressed warranties in this document and is not responsible for the products based from this document.

Table of Contents

Table of Contents i

 List of Figures i

 List of Tables i

Revision HistoryError! Bookmark not defined.

Plan ApprovalsError! Bookmark not defined.

1 Introduction 1

2 Resources used by Bridge..... 1

 2.1 Bridge installation parameters..... 1

 2.2 Shared memory allocation 2

 2.2.1 DSP configuration..... 2

 2.3 DSP MMU Page Tables 4

 2.3.1 DSP configuration..... 4

 2.4 Mailbox 4

 2.5 Host Interrupts..... 4

 2.6 DMM pool size Configuration 4

 2.7 Dynamic Heap Support 5

 2.8 Debug Print Tool support 5

 2.9 Bridge Power Management..... 5

3 Bridge Installation 6

4 DSP Memory Map 7

5 L4 Peripheral Map 9

6 Bridge Tracing 11

List of Figures

No Figures

List of Tables

Table 1 DSP Memory Map 7

Table 2 L4 Peripheral map 9

Please read the “Important Notice” on the next page.

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.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

1 Products

Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DSP	dsp.ti.com
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com

2 Applications

Audio	www.ti.com/audio
Automotive	www.ti.com/automotive
Broadband	www.ti.com/broadband
Digital Control	www.ti.com/digitalcontrol
Military	www.ti.com/military
Optical Networking	www.ti.com/opticalnetwork
Security	www.ti.com/security
Telephony	www.ti.com/telephony
Video & Imaging	www.ti.com/video
Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2008, Texas Instruments Incorporated

1 Introduction

This document describes the resources Linux bridge 3430 uses and the configurable options of bridge.

OMAP 3430 has 2 cores, ARM11 (GPP) and C64x (IVA2). The Linux OS runs on the ARM11 (GPP) core and bridge driver gets installed on this processor.

2 Resources used by Bridge

2.1 Bridge installation parameters.

It is possible to specify a number of values to parameters while installing the bridge, Below is the list of insmod parameters that can be specified.

S.No	Parameter name	Usage/example	Min value	Max Value	Default Value	Description
1	shm_size	shm_size=0x40f00	64KB	Depends on OEM configuration	512KB	Shared memory used for Bridge IPC, IVA2 MMU page tables and for IVA2 memory segments.
2	phys_mempool_base	phys_mempool_base=0x87000000	Depends on OEM configuration	Depends on OEM configuration	0	This is the starting address of the physical memory pool address reserved.
3	phys_mempool_size	pys_mempool_size=0x600000	Depends on OEM configuration	Depends on OEM configuration	0	This is the size of physical memory pool reserved.
4	dsp_inact_time	dsp_inact_time=200000	0	Any value in Milli sec.	5000	IVA2 inactivity timer value in milli seconds. When the timer expires, and if IVA2 is in inactive state, IVA2 is put into RETENTION state.
5	base_img	base_img="<base image name>	NA	NA	NULL	The dsp image to be loaded while installing the bridge.
6	GT_str	GT_str="**=01234567"	NA	NA	NULL	The GT trace levels, the numbers in the string indicate that the particular trace level is enabled and all messages with that trace level are

						printed accessing bridge.	while the
--	--	--	--	--	--	---------------------------------	--------------

2.2 Shared memory allocation

Bridge uses the kernel API calls 'dma_alloc_coherent()', 'pci_alloc_coherent()' to get physically contiguous and non-cached memory for IPC between ARM11-IVA2, and for DSP MMU pagetables.

Typically, Linux kernels have 2 to 4 MB of consistent memory pool and this pool is quickly exhausted if user loads DSP images with large external memory. To mitigate this problem, Bridge provides a mechanism to configure/pass physical memory pool at bridge installation, which would be used by bridge for all consistent memory needs (i.e IPC, DSP MMU pagetables, External memory for DSP images).

Here is an example to pass physical memory to bridge: (values are indicative only, actual values depend on OEM system configuration):

e.g: insmod -f ./bridgedriver.o phys_mempool_base=0x87000000 phys_mempool_size=0x600000
shm_size=0x40f000

2.3 Important Note on Shared Memory Configuration

Currently bridgedriver reserves 6MB memory from physical address 0x87000000 for shared memory. This memory should not be used by any other driver or Kernel.

If the "mem=" option is not set in the bootargs, kernel boots with 128 MB and the above memory is used by Kernel which results in Kernel panic in bridge initialization.

For bridgedriver to work without any issues bootargs must contain "mem=" option and the value should not be greater than 112M. The above addresses are indicative only and the original values may depend on the OEM configuration.

2.3.1 DSP configuration

Bridge allocates shared memory for GPP-DSP Inter-processor communications and for DSP external memory purposes.

Users/OEM can adjust the driver parameter 'shm_size' to meet their DSP configurations. Typically the size specified in 'shm_size' should atleast be equal to the aggregate of the following DSP segments:

'SHMMEM', 'SHMSEG0', 'SHMSEG0_GPP', 'BRIDGE_TRACE', 'DYNEXTMEM' and 'EXTMEM'.

By default, bridge allocates 1 Meg shared memory for DSP. It is possible to specify the shared memory bridge allocates through a parameter to the 'insmod' command.

Eg:

For linux 2.6 systems

Insmod ./bridgedriver.ko shm_size=0x260000

(bridge allocates 0x260000 bytes shared memory for DSP)

For linux 2.4 systems

insmod -f ./bridgedriver.o shm_size=0x260000

(bridge allocates 0x260000 bytes shared memory for DSP)

If user want to allocate shared memory of sizes more than 2MB the following parameters need to be passed to insmod.

Eg:

insmod -f ./bridgedriver.o phys_mempool_base=0x87000000 phys_mempool_size=0x600000
shm_size=0x40f000

Above values are indicative only and actual values depend on OEM system configuration.

2.4 DSP MMU Page Tables

Bridge allocates physically contiguous memory for populating the page table entries used by the DSP MMU Table walking logic.

2.4.1 DSP configuration

Bridge allocates 4KB for L1 pages. L2 page allocation is dependent on the DMM pool size configuration.

2.5 Mailbox

DSP Bridge uses two shared mailboxes for ARM/DSP inter-processor communication.

Mail Box#	Direction
0	ARM->DSP
1	DSP->ARM

2.6 Host Interrupts

Bridge attaches an ISR routine to GPP Interrupt 26 to field DSP Mail box interrupts. Bridge provides handlers for DSP MMU exceptions and notification service to registered clients. Bridge attaches an ISR routine to GPP interrupt 28 to field DSP MMU events. Below table provides the information about the interrupts.

IRQ	Source	Description
M_IRQ_26	MAIL_U0_MPU_IRQ	Mailbox user 0 interrupt request
M_IRQ_28	IVA2_MMU_IRQ	IVA2 MMU Interrupt

2.7 DMM pool size Configuration

The DMM pool size is a built in constant and OEM's can configure the DMM pool size meeting their usage requirements. OEM should ensure to have sufficient physical memory available for larger DMM pool sizes. For more information about the DMMPOOLSIZE please refer to DSP/BIOS Bridge Programming Guide.

Ex:

```
#define DMMPOOLSIZE    0x3000000
```

2.8 Dynamic Heap Support

Linux Bridge provides support for Dynamic Heap for DSP Node.

Bridge allocates node's heap in user space memory i.e in calling process context and maps the memory to DSP address space. It is responsibility of the OEMs to ensure that sufficient Physical memory is allocated for Linux to cover worst case memory allocations.

It is recommended to use 64MB system memory to support Dynamic Heap of sizes 12MB or more.

Below sample bootargs shows how to pass 64M system memory to Monta Vista® kernel while booting.

Eg :

```
bootargs=mem=64M console=ttyS0,115200n8 noinitrd root=/dev/nfs rw nfsroot=<nfsroot>, nolock, vers=2,
rsize=1024, wsize=1024 ip=dhcp devfs=mount
```

Bridge currently supports up to 16 profiles, each profile can be used for different heap size.

uProfileId for a particular node should be provided as an input while allocating a DSP Node. It is recommended to assign 0xff to DSPNODEATTRIN structure member uProfileId, if the user does not want to use this feature.

2.9 Debug Print Tool support

Default bridge builds support printing DSP debug messages on the linux console window. To enable the DSP debug trace, DSP builds need to incorporate the following statements in their 'tcf' files:

```
bridge.SYS.TRACESIZE = 0x1000; //optional
bridge.SYS.PUTCFXN = prog.extern("USR_doPutcNotifyGPP");
```

2.10 Bridge Power Management

Default Linux bridge build enables power management and DFVS. By defining the below flag at compile time disables the power management support.

DISABLE_BRIDGE_PM

3 Bridge Installation

Bridge is installed as a kernel module. Shared Memory size can be configured while installing the bridge. Values are indicative only actual values depend on OEM system configuration.

Eg:

```
insmod -f ./bridgedriver.o phys_mempool_base=0x87000000 phys_mempool_size=0x600000  
shm_size=0x40f000
```

The above command installs bridge with shared memory size 0x40f000

4 DSP Memory Map

Bridge configures the DSP memory space as per the mapping shown in Table 1. The grey rows depict memory configured for no caching. For details on DSP memory descriptions, readers are encouraged to refer to the OMAP3430 Technical Reference Manual.

Shaded address spaces in the below table are uncached.

Table 1 DSP Memory Map

Address space name	Start address (byte address)	End address (byte address)	Size (Kb)	Description
SHMMEMCHNL	0x11000000 (_SHM_BEG)	0x1100805F(_SHM_END)	32.1	Bridge shared memory for DSP-GPP communications
SHMMEMMSG	0x11008060(_MSG_BEG)	0x1100817F (_MSG_END)	0.3	Bridge shared memory for DSP-GPP communications
SHMSEG0	0x11009000 (_SHM0_BEG)	0x11009FFF	4	User shared memory segment
SHMSEG0-GPP	0x1100A000	0x1100AFFF (_SHM0_END)	4	User shared memory segment
BRIDGE_TRACE	0x1100B000 (_BRIDGE_TRACE_BEGIN)	0x1100FFFF (_BRIDGE_TRACE_END)	20	Trace Buffer for DSP Print tool
L4 peripherals	0x11800000	0x11FFFFFF	8192	Mapped to L4 peripherals, Memory and IO access possible. See table 2 for details.
DYNEXTMEM	0x20000000 (_DYN_EXT_BEG)	0x2000FFFF (_DYN_EXT_END)	64	Space for Dynamic loading of code/data. Actual size depends on TCONF/CDB configuration.

EXTMEM	0x20010000 (_EXT_BEG)	0x203EFFFF (_EXT_END)	3968	Space for External program. Actual size depends on TCONF/CDB configuration.
DMM	0x200D0000 (_EXT_END+1)	0x220D0000 (_EXT_END+1+DMMPOOLSIZE)	32768	Bridge uses this space for Dynamic memory mapping

5 L4 Peripheral Map

Following list of L4 peripherals are mapped to bridge.

Table 2 L4 Peripheral map

Peripheral NAME	Physical address	size	dsp – bytes address
PRCM	0x49006000	4KBytes	0x1181C000
GPIO1	0x48310000	4KBytes	0x11809000
GPIO2	0x49050000	4KBytes	0x1180A000
GPIO3	0x49052000	4KBytes	0x1180B000
GPIO4	0x49054000	4KBytes	0x1180C000
GPIO5	0x49056000	4KBytes	0x1180D000
IWA2WDTimer	0x49030000	4KBytes	0x1180E000
Display SS	0x48050000	4KBytes	0x1180F000
SSI Top	0x48058000	4KBytes	0x11804000
SSI GDD	0x48059000	4KBytes	0x11805000
SSI Port1	0x4805A000	4KBytes	0x11806000
SSI Port2	0x4805B000	4KBytes	0x11807000
XTI	0x48068000	4KBytes	0x11810000
UART1	0x4806A000	4KBytes	0x11811000
UART2	0x4806C000	4KBytes	0x11812000
UART3	0x49020000	4KBytes	0x11813000
McBSP1	0x48074000	4KBytes	0x11814000

McBSP2	0x49022000	4KBytes	0x11815000
McBSP3	0x49024000	4KBytes	0x11816000
McBSP4	0x49026000	4KBytes	0x11817000
McBSP5	0x48096000	4KBytes	0x11818000
GPT5	0x49038000	4KBytes	0x11800000
GPT6	0x4903a000	4KBytes	0x11801000
GPT7	0x4903c000	4KBytes	0x11802000
GPT8	0x4903e000	4KBytes	0x11803000
MailBox	0x48094000	4KBytes	0x11808000
Camera core	0x480BC000	4KBytes	0x11819000
SPI1	0x48098000	4KBytes	0x1181A000
SPI2	0x4809A000	4KBytes	0x1181B000
SDMA	0x48056000	4KBytes	0x1181d000

6 Bridge Tracing

It is possible to turn-on the tracing in bridge, through a parameter during installation of the bridge.

Eg:

```
Insmod ./bridgedrive.ko GT_str = "***=567"
```

The value of "Trace" is a mask name/value pair. The name in the default case is "*" - meaning apply to all mask names. The mask is, by default "1567" - meaning turn on only levels 1, 5, 6, and 7. If you change the value of "Trace" to "*=01234567", you'll enable full tracing (which will result in a lot of output), but this is often useful in identifying what went wrong internally.

You can also configure "Trace" to set specific masks to specific levels during debugging. To configure more than one mask, you can separate masks with a semi-colon. Below is a list of mask names.

For example, to use default setting for all modules, except memory (which you don't want to see), and nodes (which you want to see *everything*), you could set "Trace" to the following:

```
"*=1567;MM=;NO=01234567"
```

Mask Name	Module Prefix	Module Description
CF	CFG	Config
CM	CMM	Comm Memory Manager
CO	COD	Code Manager
CS	CSL	C Standard Library
DI	DISP	Dispatcher
DM	DMM	Dynamic Memory Mapper
DP	DPC	Deferred Procedure Call
DS	DSP	High-level "DSP" interface
DV	DEV	Device
IO	IO	IO
IS	ISR	Interrupt Service Routine
IV	IVA	IVA
KF	KFILE	File routines(used in kernel mode)
LD	LDR	Loader
LS	LST	List
MG	MGR	Manager
MM	MEM	Memory

MS	MSG	Messaging
NO	NODE	Node
NY	NTFY	Notification
OS	OSAL	OS Abstraction Layer
PF	PERF	Performance
PR	PROC	Processor
PR	PRCS	Process
RG	REG	Registration
UL	UTIL	Utilities
ST	STRM	Stream
SY	SYNC	Synchronization
WD	WMD	Minidriver