- **Multi-instantiable and re-entrant safe driver**
- **Modeled after TI Device Driver Architecture for Stream Class Devices that allows for easy porting and customization**

Description

This EDMA3 Resource Manager has been developed for the following h/w and s/w environment

Hardware:

- Target Boards: DA830 EVM, TCI6608 (Little and Big Endian) Simulator, TCI6616 (Little and Big Endian) Simulator, C6748 EVM, OMAPL138 EVM, C6472 (Little and Big Endian) EVM, TCI6486 (Little and Big Endian) EVM, TI816X EVM, TI814X EVM, TI811X EVM, C6670 EVM,C6678 EVM, tci6614 EVM , tci6614 Simulator, Vayu EVM (DRA7xx), TDA3xx, DRA72x.
- Emulation Setup: TI Emulator
- Cabling: Standard Serial Connector

Software:

- CCS: 5.5.0
- Operating System: SYS BIOS 6.40.03.39
- C6x Code generation tools: 7.4.4
- TMS470 Code Generation Tools: 5.1.5
- Arm GCC A15 Toolchain: gcc-arm-none-eabi-4_7-2013q3
- ARP32 Code Generation Tools: 1.0.2
- XDC Tool chain: 3.30.04.52
- TCI6608/TCI6616 Simulator: 1.0.0

Other S/W Components Used:

- None

Capabilities

The DSP BIOS EDMA3 Resource Manager adopts a scalable architecture that eases customization/extension

- Isolates H/W and OS Accesses, Easy to maintain & re-target to new platforms
- Can stack custom-functions along control/data-path to realize “driver filters”
- Supports Multiple Instances

For easy and quick reconfiguration of driver for different SoCs, all its global configurable parameters can be passed at run time to the API EDMA3_DRV_create (), to create the SoC specific EDMA3 Driver Object. In case this configuration is not passed at run time, it can be taken from the EDMA3 configuration file edma3_<PLATFORM_NAME>_cfg.c, for the specific SoC, if it has been provided there. The configuration files can be found in “edma3_lld_<VERSION_NUMBER>\packages\ti\sdo\edma3\rm\src\configs” folder.

Similarly, the shadow region specific information can also be passed at run time to the API EDMA3_DRV_open (), to create region specific EDMA3 Driver Instance. In case this configuration is not passed at run time, it can be taken from the EDMA3 configuration file edma3_<PLATFORM_NAME>_cfg.c, for the specific SoC, if it has been provided there. The configuration files can be found in “edma3_lld_<VERSION_NUMBER>\packages\ti\sdo\edma3\rm\src\configs” folder.

EDMA3 Resource Manager APIs:

EDMA3_RM_create ()	Create (and initialize) a given EDMA3 Resource Manager (object)
EDMA3_RM_delete ()	Delete a given EDMA3 Resource Manager (object)
EDMA3_RM_open ()	Open instance of the EDMA3 Resource Manager
EDMA3_RM_close ()	Close instance of the EDMA3 Resource Manager
EDMA3_RM_allocResource ()	Request for a resource
EDMA3_RM_freeResource ()	Free the earlier requested resource
EDMA3_RM_mapEdmaChannel ()	Bind a DMA Channel to a PaRAM Set
EDMA3_RM_mapQdmaChannel ()	Bind a QDMA Channel to a PaRAM Set and set the trigger word for the QDMA channel
EDMA3_RM_registerTccCb ()	Register a callback against a TCC
EDMA3_RM_unregisterTccCb ()	Unregister the callback from the TCC
EDMA3_RM_allocContiguousResource ()	Allocate a contiguous region of specified resource
EDMA3_RM_freeContiguousResource ()	Free a contiguous region of specified resource
EDMA3_RM_setCCRegister ()	Set the Channel Controller (CC) Register value, by specifying the register offset and the new value.

EDMA3_RM_getCCRegister ()	Get the Channel Controller (CC) Register value, by specifying the register offset.
EDMA3_RM_waitAndClearTcc ()	Wait for a transfer completion interrupt to occur on the specific TCC.
EDMA3_RM_checkAndClearTcc ()	Returns the status of a previously initiated transfer.
EDMA3_RM_allocLogicalChannel ()	Request a DMA/QDMA/Link channel.
EDMA3_RM_freeLogicalChannel ()	This API is used to free the specified channel (DMA/QDMA/Link) and its associated resources (PaRAM Set, TCC etc).
EDMA3_RM_setPaRAM ()	Set the PaRAM Set associated with a logical channel
EDMA3_RM_getPaRAM ()	Get the PaRAM Set associated with a logical channel
EDMA3_RM_getPaRAMPhyAddr ()	Get the PaRAM Set Physical Address associated with a logical channel.
EDMA3_RM_getBaseAddress ()	Get the Channel Controller or Transfer Controller (n) Physical Address.
EDMA3_RM_getGblConfigParams ()	Get the SoC specific configuration structure for the EDMA3 Hardware.
EDMA3_RM_getInstanceInitCfg ()	Get the Resource Manager Instance specific configuration structure for different EDMA3 resources' usage (owned resources, reserved resources etc).
EDMA3_RM_Ioctl ()	This function provides IOCTL functionality for EDMA3 Resource Manager.
EDMA3_RM_initXbarEventMap ()	This function provides the functionality to map cross bar event to EDMA channels.

EDMA3 Resource Manager Performance Characteristics

S No	Platform	Code	Data (In bytes)		Total
		(In bytes)	Initialized	Uninitialized	(In bytes)
1	C6472	25920	2723	31239	59882
2	C6748	25920	3451	33367	62738
3	DA830	25920	2051	30567	58538
4	TI816X	25920	2723	31239	59882
5	OMAPL138	25920	3451	33367	62738
6	TCI6486	25920	2723	31239	59882
7	TCI6498	25920	6867	35383	68170

References

[1] EDMA3 Module Hardware Specifications

[2] DSP BIOS Documentation

[3] EDMA3 Resource Manager Documentation

Glossary

PaRAM Set Parameter RAM Set in an EDMA3 Controller

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.

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

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

LICENSE