**SDK**

**Seeing Prints from M3 on Linux using UIA in SDK**

**Abstract**

UIA enables seeing prints from M3 on a Linux terminal. However, enabling UIA in SDK requires re-building the M3 binaries with few changes in makefiles & xdc config files. This document describes the steps to enable UIA in SDK. VC3 has been taken as an example to demonstrate this. The same applies to VS2 as well. Note that this document is not a comprehensive demonstration of the capabilities of UIA in the context of SDK. This document is specifically targeted to enable users of SDK with access to M3 source code to see prints from M3 on a linux terminal.

***The document assumes SDK release version 5.0.0.11 and UIA version 1.00.01.05.***

***Note that currently UIA package is not part of SDK release package but will be available in future SDK releases.***

Download public DM816x SDK to access UIA v 1.00.01.05 from:

<http://software-dl.ti.com/dsps/dsps_public_sw/ezsdk/latest/index_FDS.html>

and copy into EXTERNAL\_SW\_ROOT directory of SDK 5.0.0.11.

Please refer the LoggerSM documentation (part of UIA) for exact details on this module.

**Changes in Makefile and XDC Config files**

• Install UIA version 1.00.01.05 under the $(EXTERNAL\_SW\_ROOT) folder where all other foundation software is available \makerules\env.mk: This already has the following variables defined:

uia\_PATH =$(EXTERNAL\_SW\_ROOT)/uia\_1\_00\_01\_05

uia\_INCLUDE = $(uia\_PATH)/packages

• Add the following in \ti\omx\demos\vc3\VpsAppMain.cfg

/\* ================ Loggerconfiguration ================ \*/

var Defaults =xdc.useModule('xdc.runtime.Defaults');

/\* Create a LoggerSM instance and use it for all logging. Make

sure it is at the same section for all cores (DSP,Video, VPSS) and not

put anything else in "LOGGERSM". All cores will share this same memory.

All cores must have numCores and sharedMemorySize be the same value.

Note: LOGGERSM memory segment is defined in

ti/uia/examples/evmTI816X/video/Platform.xdc \*/

var LoggerSM = xdc.useModule('ti.uia.runtime.LoggerSM');

if(cfgArgs.coreName.match("VIDEO-M3")) {

LoggerSM.partitionId = 1;

} else {

LoggerSM.partitionId = 2;

}

LoggerSM.bufSection = ".loggerSM";

LoggerSM.sharedMemorySize = 0x00020000;

LoggerSM.numPartitions =3;

Defaults.common$.logger = LoggerSM.create();

LoggerSM.decode = true;

LoggerSM.overwrite = false;

/\* Make sure the section is in LOGGERSM (defined in ti\uia\examples\platforms\evmti816x). Also make

sure it is a NOLOAD section. This avoids wiping out another cores logger memory when more than one

cores is loaded \*/

Program.sectMap[".loggerSM"] = new Program.SectionSpec(); Program.sectMap[".loggerSM"].loadSegment ="LOGGERSM";

Program.sectMap[".loggerSM"].type = "NOLOAD";

/\* Turn on ANALYSIS for benchmark events and others for Log\_print used in app \*/

Main.common$.diags\_ENTRY = Diags.ALWAYS\_ON;

Main.common$.diags\_EXIT = Diags.ALWAYS\_ON;

Main.common$.diags\_LIFECYCLE = Diags.ALWAYS\_ON;

Main.common$.diags\_INTERNAL =Diags.ALWAYS\_ON;

Main.common$.diags\_ASSERT = Diags.ALWAYS\_ON;

Main.common$.diags\_STATUS =Diags.ALWAYS\_ON;

Main.common$.diags\_USER1 = Diags.ALWAYS\_ON;

Main.common$.diags\_USER2 =Diags.ALWAYS\_ON;

• Note that LOGGERSM has been configured to associate Video-M3 with partitionId 1 and VPSS-M3 with

partitionId 2 (It is assumed that DSP has partitionId 0)

if (cfgArgs.coreName.match("VIDEO-M3")) {

LoggerSM.partitionId = 1; }

else {

LogerSM.partitionId = 2; }

• LOGGERSM section needs to be defined in \ti\omx\build\MemSegmentDefinition.xs. Note that this section needs to be in non-cacheable DDR space & the same memory segment needs to be shared between the two M3 cores – Video-M3 & VPSS-M3.

As an example, reduce memory[22] in MemorySegmentDefinition.xs and add memory[24] as shown below.

The non-cacheable DDR region used by both M3 cores in this case as defined by LOGGERSM is 0xB2C00000.

\ti\omx\demos\vc3\AmmuCfg.cfg defines this DDR region as non-cacheable

memory[22] = ["FBDEV\_V4L2\_Mem",

{

name: "FBDEV\_V4L2\_Mem",

base: 0xB2D00000,

len: 0x01000000,

space: "data"

}];

memory[23] = ["FQMEM\_BUFFERS\_NON\_CACHED",

{

name: "FQMEM\_BUFFERS\_NON\_CACHED",

base: 0xB3D00000,

len: 0x0C300000,

space: "data"

}];

/\* For UIA logging to linux terminal \*/

memory[24] = ["LOGGERSM",

{

name: "LOGGERSM",

base: 0xB2C00000,

len: 0x00020000,

space: "data"

}];

**Running VC3**

Note down the IP address of the EVM before running VC3. In this example, the IP address of the EVM is

172.24.190.7. The steps to run VC3 are the same as before.

**Building loggerSMDump.out executable from UIA examples**

The loggerSMDump utility available at $(uia\_PATH)/packages/ti/uia/examples/evmti816x pulls out logs from the loggerSM buffer as allocated above and prints the logs on the linux terminal. By default, loggerSMDump utility is configured for 3 cores on DM816x, i.e, DSP, Video-M3 & VPSS-M3.

• Compile loggerSMDump.out using the Makefile at $(uia\_PATH)/packages/ti/uia/examples/evmti816x

• Copy the loggerSMDump.out executable to the same location as the vc3 executables

**Viewing Prints on a Linux Terminal**

• After running the VC3 executable, open a telnet session using the IP address of the EVM telnet

172.24.190.7

• Change to the folder which has the loggerSMDump.out executable

• Run the following

command – ./loggerSMDump.out 0xB2C00000 all

• Note that the address should match the one allocated through changes in \ti\omx\build\MemSegmentDefinition.xs (non-cacheable DDR region)

• Execute ./loggerSMDump.out to see the various options available in this utility.

**Example Prints from loggerSMDump.out**

Given below is an example of the logs coming from the two M3 cores. These prints are enabled on M3 as a result of the build settings in the VpsAppMain.cfg where the diags masks for xdc.runtime.Main are enabled. The user may opt to turn off some of these masks to limit prints coming from M3 core. Additionally, those having access to M3 source code may add more Log\_prints to enable more prints from M3 cores. The format of print is

– N:*CoreName* P:*PartitionId* #:*SequenceId* T:*64bit timestamp* S:*Print\_from\_slave\_core*

Seeing Prints from M3 on Linux using UIA in SDK 3 Example –

N:Video P:1 #:44119 T:0000000f|287f5c19 S:IvaScheduler\_getIvahdId::Line 72::algHandle 0x8f0bd1c0::allocated IVAHD 0

N:VPSS P:2 #:05771 T:0000000f|309ea647 S:FVID\_DEQUEUE\_INFO:: OmxPortIndex:6 ,

FvidFramePtr :@97382b34 , FvidFrameBufPtr : @b47b5480 , OmxHdr:@97383110 ,

FQHdr: @9a00e080