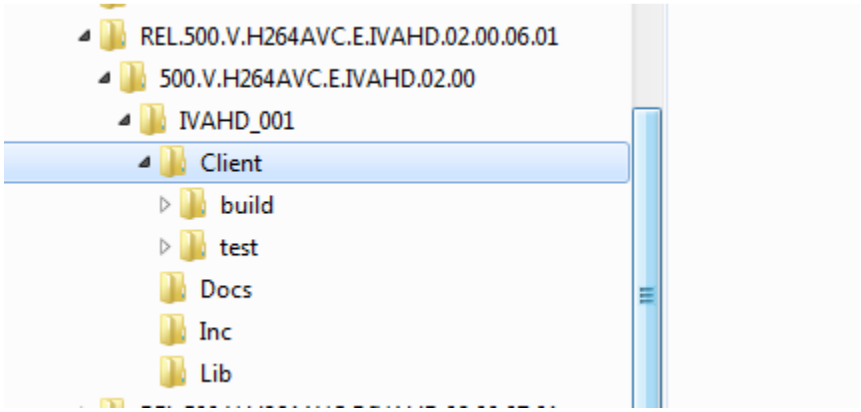
如何离线方式编译和测试 codec

客户在实际使用中有时候需要静态分析 codec 编码的数据和结果。TI 的 RDK 中 release 的 codec 包含文件到文件方式的测试用例。用文件到文件的方式配合不同的编码参数配置文件，能够方便用户分析和测试实际开发中同 codec 相关的一些问题。在 TI 的 RDK 中，codec 库文件通常存放在/ti_tools/codecs/ 下面，不同的硬件版本和 RDK 版本有稍许不同，例如 DM8127 的 RDK 就存放在/Source/ti_tools/codecs-dm814x 目录下面。在该目录中会包含 H.264 , MPEG4, MJPEG 等编码和解码的库，测试用例，release note，测试报告和使用指南。如下图就是我们 H.264 编码库的一个目录结构。建议大家先浏览一下 Docs 目录中的使用指南和测试报告。



下面我们以 H.264 解码库为例来介绍如何编译和测试 codec。我们以 IPNC RDK3.8 为例， 在 RDK 的 IVAHD_001/Lib 目录下有编码器的库文件，而对应的测试程序在 IVAHD_001/test 目录下面。软件所有的编译工作在 windows 操作系统完成。但是工具链和库文件需要从安装 RDK 的 linux 机器拷贝到 windows 工作目录。我们以 C:\test 目录作为工作目录。

第一步：将 codec 的完整目录拷贝到工作目录中

将 ti_tools\ivahd_hdvicp20api_01_00_00_23 的完整目录拷贝到工作目录中。注意不同的 codec 对应的 ivahd_hdvicp20api 版本有微小不同，建议直接使用 RDK 中包含的 hdvicp20api 库文件。

第二步：release note 中给出了 HDVICP 开发的工具链和环境要求，里面详细列出了当前版本开发需要的软件列表。由于我们用的是 HDVICP 的库，所以不必要下载所有的工具链，只需要下载 xdctools，framework component 两项就可以。

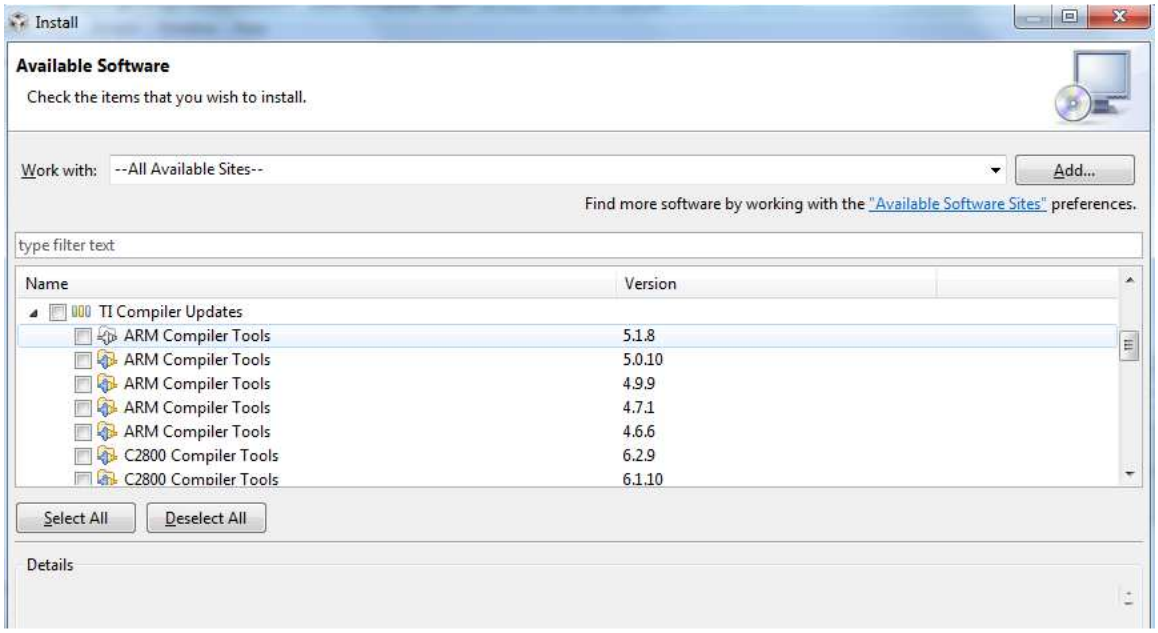
Framework Component version 3.20.00.22

（http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/fc/3_20_00_22/index_FDS.html ）

XDC tools version 3.20.04.68
(http://downloads.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/rtsc/)

由于产生HDVICP2的编译工具 cgt_TMS470 没有提供下载，需要安装CCS开发环境然后用CCS开发环境中的编译工具。该文档是基于CCS5.5 开发环境进行的测试。下面给出了CCS5.5安装编译工具的方法。
（http://processors.wiki.ti.com/index.php/Download_CCS）

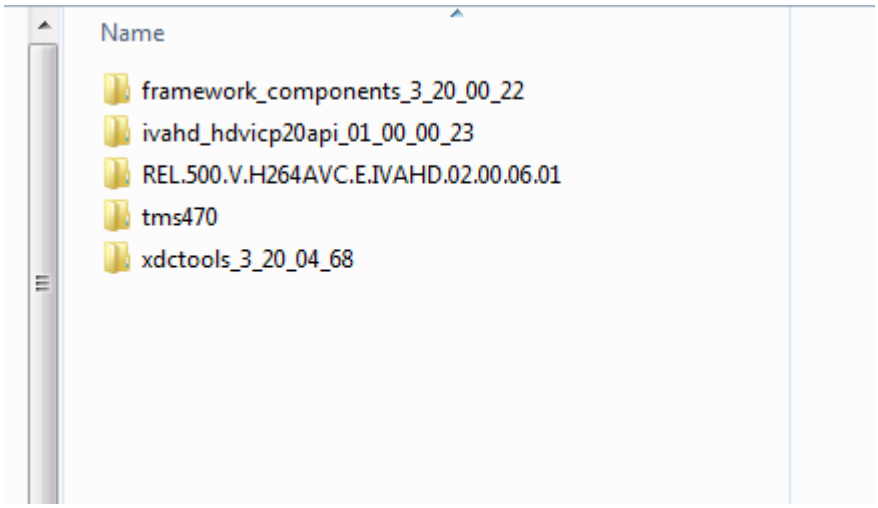
安装完成CCS后，有可能ARM compile tool 并没有安装对应的版本。在CCS 环境中 help -> install new software … 中可以通过在线的方式安装ARM compiler tools。



将CCS中的工具链（C:\ti\ccsv5\tools\compiler\arm_5.1.8）拷贝到工作目录中，并改名为tms470

编译器需要用到 gmake，在 xdctool 安装目录下有，一个简单的方法就是将当前版本的 gmake 拷贝到工作目录中。

做完上述工作后的工作目录如下图



第三步：修改的 make 文件。按照各个工具和源代码所放置的位置修改 \IVAHD_001\Client\build\TestAppDM816x\make 下的 makefile

将下面几行修改为工作目录中的工具所在的位置

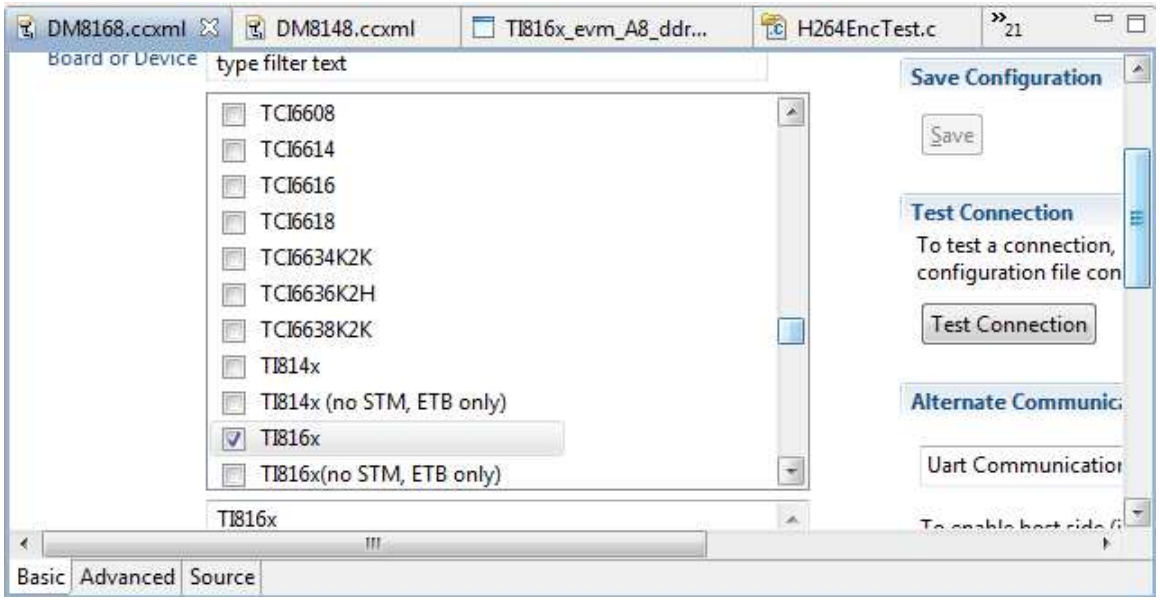
HDVICP_DIR = %HDVICP2_INSTALL_DIR%
FC_DIR = %FC_INSTALL_DIR%
CG_TOOL = %CG_TOOL_DIR_M3%
XDC_DIR = %XDCROOT%

改完后的结果如下：
HDVICP_DIR =
C:\test\ivahd_hdvicp20api_01_00_00_23\packages\ti\sdo\codecs\hdvicp20api
FC_DIR = C:\test\framework_components_3_20_00_22
CG_TOOL = C:\test\tms470
XDC_DIR = C:\test\xdctools_3_20_04_68

第四步：在 makefile 文件所在目录下运行 gmake

Client\Build\TestAppDM816x\make> gmake -s deps
Client\Build\TestAppDM816x\make> gmake -k -s all

第五步：在评估板上测试 codec
本次实验中采用的是 DM8168 EVM 板，

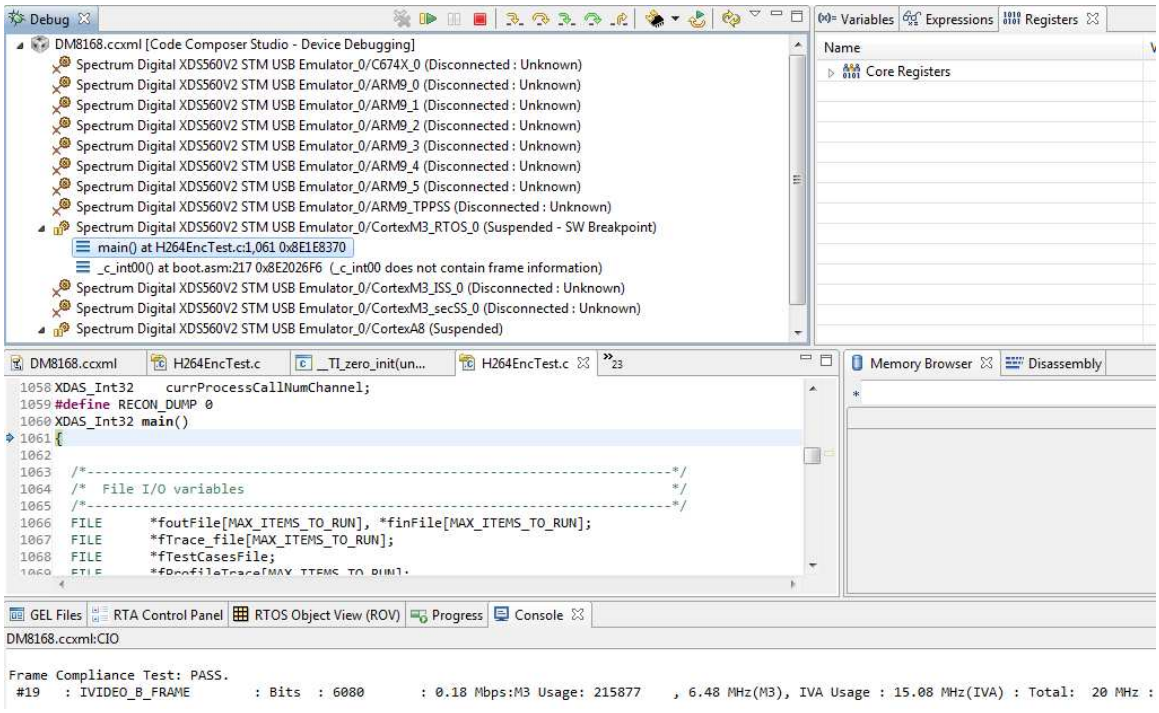


评估板对应的 GEL 文件用的是 wiki 中的对应文件。
http://processors.wiki.ti.com/index.php/File:TI816x_ddr3.zip

在测试的时候由于 DM8168 是没有操作系统在运行，因此 M3 和 HDVICP 都需要通过 GEL 文件来使能才能够正常运行编译出来的.out 文件。请按照下列顺序执行 GEL 脚本

DM816x PLL -> DM816xMainPLL
DM816x External Memories -> DDR3_796MHz_doall
DM816x OMX init -> MediaController
DM816x OMX init -> CortexM3_0
DM816x OMX init -> CortexM3_1
DM816x OMX init -> IVAHD0
DM816x OMX init-> MediaControllerClkEnable

连接上 CortexM3_RTOS_0 这个核后下载.out 文件，直接运行即可。



以上是编译和测试 codec 的方法，希望对大家实际工作有所帮助。