```
/*
 *  ======== config.bld ========
 *  Build configuration script for HDVPSS drivers
 */

/* load the required modules for the configuration */

var M3 = xdc.useModule('ti.targets.arm.elf.M3');
var C674 = xdc.useModule('ti.targets.elf.C674');

var buildReleaseConfig = true;

/* configure  the options for the M3 targets     */

/* M3 compiler directory path                    */
M3.rootDir = java.lang.System.getenv("CGTOOLS");

/* linker options */

M3.lnkOpts.suffix += " --zero_init=off ";
M3.lnkOpts.suffix += " --dynamic --retain=_Ipc_ResetVector";

/* compiler options                                */
M3.ccOpts.suffix += " --gcc -DTI_816X_BUILD -DPLATFORM_EVM_SI -DSYSLINK_BUILD_RTOS -
DUSE_SYSLINK_NOTIFY=0 -DUTILS_ASSERT_ENABLE";

/* set default platform and list of all interested
 * platforms for M3
 */
M3.platforms = [
                        "ti.platforms.evmTI816X:core0",
                        "ti.platforms.evmTI816X:core1",
               ];

/* Select the default platform
 *
 * Making core1 as defualt core configuration to be used
 *  Core 0 ==    Ducati.M3.VIDEO
 *  Core 1 ==    Ducati.M3.VPS
 */
M3.platform = M3.platforms[1];

/* configure  the options for the C674 targets     */

/* C674 compiler directory path                    */
C674.rootDir = java.lang.System.getenv("CGTOOLS_DSP");

/* linker options */

C674.lnkOpts.suffix += " --zero_init=off ";
C674.lnkOpts.suffix += " --dynamic --retain=_Ipc_ResetVector";

/* compiler options                                */
C674.ccOpts.suffix += " -DTI_816X_BUILD -DPLATFORM_EVM_SI -DSYSLINK_BUILD_RTOS -
DUSE_SYSLINK_NOTIFY=0";

C674.platforms = ["ti.platforms.evmTI816X:plat"];
C674.platform = C674.platforms[0];

/* list interested targets in Build.targets array  */
```

```
Build.targets = [
                    M3,
                    C674,
                ];

var KB=1024;
var MB=KB*KB;

var DDR3_ADDR;
var DDR3_SIZE;

var OCMC0_ADDR;
var OCMC1_ADDR;
var OCMC_SIZE;

var LINUX_ADDR;
var LINUX_SIZE;

var SR0_ADDR;
var SR0_SIZE;

var SR1_ADDR;
var SR1_SIZE;

var SR3_INTRADUCATI_IPC_ADDR;
var SR3_INTRADUCATI_IPC_SIZE;

var VIDEO_M3_CODE_ADDR;
var VIDEO_M3_CODE_SIZE;

var VIDEO_M3_DATA_ADDR;
var VIDEO_M3_DATA_SIZE;

var SR2_FRAME_BUFFER_ADDR;
var SR2_FRAME_BUFFER_SIZE;

var DSS_M3_CODE_ADDR;
var DSS_M3_CODE_SIZE;

var DSS_M3_DATA_ADDR;
var DSS_M3_DATA_SIZE;

var DSP_CODE_ADDR;
var DSP_CODE_SIZE;

var DSP_M3_DATA_ADDR;
var DSP_M3_DATA_SIZE;

var TILER_ADDR;
var TILER_SIZE;

var HDVPSS_DESC_ADDR;
var HDVPSS_DESC_SIZE;

var HDVPSS_SHARED_ADDR;
var HDVPSS_SHARED_SIZE;

var NOTIFY_SHARED_ADDR;
var NOTIFY_SHARED_SIZE;
```

```
var REMOTE_DEBUG_ADDR;
var REMOTE_DEBUG_SIZE;

DDR3_ADDR                  = 0x80000000;
DDR3_SIZE                  = 2048*MB;

OCMC0_ADDR                 = 0x40300000;
OCMC1_ADDR                 = 0x40400000;
OCMC0_RUN_ADDR             = 0x00300000;
OCMC1_RUN_ADDR             = 0x00400000;
OCMC_SIZE                  = 256*KB;

L2_SRAM_ADDR               = 0x55024000;
L2_SRAM_SIZE               = 128*KB;
L2_SRAM_RUN_ADDR           = 0x20004000;

DUCATI_WB_WA_ADDR          = 0x20000000;

/* first 512MB */
LINUX_SIZE                 = 128*MB;
SR1_SIZE                   = 336*MB;
SR3_INTRADUCATI_IPC_SIZE   =          124*KB;
VIDEO_M3_CODE_SIZE         =  2*MB  + 512*KB;
VIDEO_M3_BSS_SIZE          =  9*MB  + 512*KB;
VIDEO_M3_DATA_SIZE         =  2*MB  + 512*KB;
DSS_M3_CODE_SIZE           =  1*MB  + 512*KB;
DSS_M3_BSS_SIZE            = 11*MB  + 512*KB;
DSS_M3_DATA_SIZE           =  5*MB  + 512*KB;
DSP_CODE_SIZE              =  1*MB;
DSP_DATA_SIZE              = 13*MB  + 900*KB;



/* second 512MB */
/* Tiler Buffers in the bottom 512MB */
TILER_SIZE                  = 256*MB; /* (128+128) - MUST be aligned on 128MB boundary */
SR2_FRAME_BUFFER_SIZE       = 234*MB - 256*KB;
SR0_SIZE                    = 15*MB;
VIDEO_M3_EXCEPTION_CTX_SIZE = 128*KB;
VPSS_M3_EXCEPTION_CTX_SIZE  = 128*KB;
HDVPSS_DESC_SIZE            = 2*MB;
HDVPSS_SHARED_SIZE          = 2*MB;
NOTIFY_SHARED_SIZE          = 2*MB;
REMOTE_DEBUG_SIZE           = 1*MB;

/* third and fourth 512 MB */
LINUX_SIZE_SEGMENT2        = 1024*MB;
print ("Memory Map - 2GB DDR, > 1GB Linux in 2 segments");

print ("  0x80000000     +--------------------+");
print ("         ^       |                    |");
print ("         |       |  " + (LINUX_SIZE / MB) + " MB           | Linux");
print ("         |       |                    |");
print ("         |       +--------------------+");
print ("         |       |  " + (SR1_SIZE / MB) + "MB          | (SR1) Bitstream buffer");
print ("         |       |                    | Cached on A8. Cached on M3, although access
by DMAs");
print ("         |       +--------------------+ ");
print ("         |       |   " + (SR3_INTRADUCATI_IPC_SIZE / KB) + " KB          |
(SR3)InterDucati IPC ListMP .Cached on M3 ");
```

```
print ("         |       +-------------------+  ");
print ("         +       |   " + (VIDEO_M3_CODE_SIZE / MB) + " MB          | Video M3
Code");
print ("       512 MB    +-------------------+");
print ("         +       |   " + (VIDEO_M3_BSS_SIZE / MB) + " MB         | Video M3 BSS");
print ("         |       +-------------------+");
print ("         |       |   " + (VIDEO_M3_DATA_SIZE / MB) + " MB          | Video  M3
Data");
print ("         |       +-------------------+");
print ("         |       |   " + (DSS_M3_CODE_SIZE/ MB) + " MB          | VPSS  M3 Code");
print ("         |       +-------------------+");
print ("         |       |   " + (DSS_M3_BSS_SIZE/ MB) + " MB         | VPSS  M3 BSS");
print ("         |       +-------------------+");
print ("         |       |   " + (DSS_M3_DATA_SIZE/ MB) + " MB           | VPSS  M3
Data");
print ("         |       +-------------------+");
print ("         |       |   " + ( DSP_CODE_SIZE / KB) + " KB          | DSP Code");
print ("         |       +-------------------+");
print ("         v       |   " + (DSP_DATA_SIZE / MB) + " MB         | DSP Data");
print ("  0xA0000000     +-------------------+");
print ("         ^       |   " + (TILER_SIZE / MB) + " MB          | Tiled 8-bit + 16-bit
region");
print ("         |       +-------------------+");
print ("         |       |  " + (SR2_FRAME_BUFFER_SIZE / MB) + " MB        | (SR2) Frame
Buffer Region - <VPSS - Video M3 Frame Buf>");
print ("         |       +-------------------+  ");
print ("         +       |                   |");
print ("       512 MB    |  " + (SR0_SIZE / MB) + " MB            | (SR0) Syslink MsgQ/IPC
List MP - <Non-cached on M3>");
print ("         +       +-------------------+");
print ("         |       |  " + (VIDEO_M3_EXCEPTION_CTX_SIZE / KB) + " KB           |
Video M3 exception context");
print ("         |       +-------------------+  ");
print ("         |       |  " + (VPSS_M3_EXCEPTION_CTX_SIZE / KB) + "KB            | Vpss
M3 exception context");
print ("         |       +-------------------+  ");
print ("         |       |  " + (HDVPSS_DESC_SIZE / MB) + " MB             | VPSS M3 -
VPDMA Descriptor");
print ("         |       +-------------------+");
print ("         |       |  " + (HDVPSS_SHARED_SIZE / MB) + " MB             | VPSS M3 -
FBDev Shared Memory");
print ("         |       +-------------------+");
print ("         |       |  " + (NOTIFY_SHARED_SIZE / MB) + " MB             | Host - VPSS
M3 Notify(For FBDev)");
print ("         |       +-------------------+");
print ("         v       |  " + (REMOTE_DEBUG_SIZE / MB) + " MB             | Remote Debug
Print");
print ("   0xC0000000    +-------------------+");
print ("         ^       |                   |");
print ("         |       |                   |");
print ("         |       |  " + (LINUX_SIZE_SEGMENT2 / MB) + " MB          | Memory for
Linux Kernel");
print ("         |       |                   |" + " Available only in split of linux
kernel");
print ("         |       |                   |");
print ("         v       |                   |");
print ("   0xFFFFFFFF    +-------------------+");


/* first 512MB */
```

```
LINUX_ADDR                  = DDR3_ADDR;
SR1_ADDR                    = LINUX_ADDR                                          +
LINUX_SIZE;
SR3_INTRADUCATI_IPC_ADDR    = SR1_ADDR                                            +
SR1_SIZE;
VIDEO_M3_CODE_ADDR          = SR3_INTRADUCATI_IPC_ADDR            +
SR3_INTRADUCATI_IPC_SIZE;
VIDEO_M3_DATA_ADDR          = VIDEO_M3_CODE_ADDR                          +
VIDEO_M3_CODE_SIZE;
VIDEO_M3_BSS_ADDR           = VIDEO_M3_DATA_ADDR                          +
VIDEO_M3_DATA_SIZE;
VIDEO_M3_BSS_MAPPED_ADDR    = (VIDEO_M3_BSS_ADDR - DDR3_ADDR) + DUCATI_WB_WA_ADDR;
DSS_M3_CODE_ADDR            = VIDEO_M3_BSS_ADDR                           +
VIDEO_M3_BSS_SIZE;
DSS_M3_DATA_ADDR            = DSS_M3_CODE_ADDR                            +
DSS_M3_CODE_SIZE;
DSS_M3_BSS_ADDR             = DSS_M3_DATA_ADDR                            +
DSS_M3_DATA_SIZE;
DSS_M3_BSS_MAPPED_ADDR      = (DSS_M3_BSS_ADDR - DDR3_ADDR)       + DUCATI_WB_WA_ADDR;
DSP_CODE_ADDR               = DSS_M3_BSS_ADDR                             +
DSS_M3_BSS_SIZE;
DSP_DATA_ADDR               = DSP_CODE_ADDR                                       +
DSP_CODE_SIZE;

/* second 512MB */
/* Tiler Buffers in the bottom 512MB */
TILER_ADDR                   = DDR3_ADDR                                        +
DDR3_SIZE/4;
SR2_FRAME_BUFFER_ADDR        = TILER_ADDR                                       +
TILER_SIZE;
VIDEO_M3_EXCEPTION_CTX_ADDR = SR2_FRAME_BUFFER_ADDR                     +
SR2_FRAME_BUFFER_SIZE;
VPSS_M3_EXCEPTION_CTX_ADDR  = VIDEO_M3_EXCEPTION_CTX_ADDR       +
VIDEO_M3_EXCEPTION_CTX_SIZE;
HDVPSS_DESC_ADDR             = VPSS_M3_EXCEPTION_CTX_ADDR        +
VPSS_M3_EXCEPTION_CTX_SIZE;
HDVPSS_SHARED_ADDR           = HDVPSS_DESC_ADDR                          +
HDVPSS_DESC_SIZE;
NOTIFY_SHARED_ADDR           = HDVPSS_SHARED_ADDR                        +
HDVPSS_SHARED_SIZE;
REMOTE_DEBUG_ADDR            = NOTIFY_SHARED_ADDR                        +
NOTIFY_SHARED_SIZE;
SR0_ADDR                     = REMOTE_DEBUG_ADDR                     + REMOTE_DEBUG_SIZE;

if ((DSP_DATA_ADDR + DSP_DATA_SIZE) > TILER_ADDR)
{
  throw xdc.$$XDCException("MEMORY_MAP OVERFLOW ERROR ",
                           "\nRegion End: " + "0x" +
java.lang.Long.toHexString(TILER_ADDR) +
                           "\nActual End: " + "0x" +
java.lang.Long.toHexString(DSP_DATA_ADDR + DSP_DATA_SIZE ));
}

if ((SR0_ADDR + SR0_SIZE) > DDR3_ADDR + DDR3_SIZE/2)
{
  throw xdc.$$XDCException("MEMORY_MAP OVERFLOW ERROR ",
                           "\nRegion End: " + "0x" + java.lang.Long.toHexString(DDR3_ADDR
+ DDR3_SIZE) +
                           "\nActual End: " + "0x" + java.lang.Long.toHexString(SR0_ADDR +
SR0_SIZE)
```

```
                                                    );
}

Build.platformTable["ti.platforms.evmTI816X:core1"] =
{
    externalMemoryMap:
    [
        ["DDR3_RAM", {
            comment: "DDR3_RAM",
            name: "DDR3_RAM",
            base: DDR3_ADDR,
            len:  DDR3_SIZE
        }],
        ["OCMC1_RAM", {
            comment: "OCMC1_RAM",
            name: "OCMC1_RAM",
            base: OCMC1_ADDR,
            len:  OCMC_SIZE
        }],
        ["VIDEO_M3_BSS_MAPPED_MEM", {
            comment : "VIDEO_M3_BSS_MAPPED_MEM",
            name    : "VIDEO_M3_BSS_MAPPED_MEM",
            base    : VIDEO_M3_BSS_MAPPED_ADDR,
            len     : VIDEO_M3_BSS_SIZE
        }],
        ["DSS_M3_BSS_MAPPED_MEM", {
            comment : "DSS_M3_BSS_MAPPED_MEM",
            name    : "DSS_M3_BSS_MAPPED_MEM",
            base    : DSS_M3_BSS_MAPPED_ADDR,
            len     : DSS_M3_BSS_SIZE
        }],
    ],
    customMemoryMap:
    [
        ["LINUX_MEM", {
            comment : "LINUX_MEM",
            name    : "LINUX_MEM",
            base    : LINUX_ADDR,
            len     : LINUX_SIZE
        }],
        ["SR1", {
            comment : "SR1",
            name    : "SR1",
            base    : SR1_ADDR,
            len     : SR1_SIZE
        }],
        ["SR3_INTRADUCATI_IPC", {
            comment : "SR3_INTRADUCATI_IPC",
            name    : "SR3_INTRADUCATI_IPC",
            base    : SR3_INTRADUCATI_IPC_ADDR,
            len     : SR3_INTRADUCATI_IPC_SIZE
        }],
        ["VIDEO_M3_CODE_MEM", {
            comment : "VIDEO_M3_CODE_MEM",
            name    : "VIDEO_M3_CODE_MEM",
            base    : VIDEO_M3_CODE_ADDR,
            len     : VIDEO_M3_CODE_SIZE
        }],
        ["VIDEO_M3_DATA_MEM", {
            comment : "VIDEO_M3_DATA_MEM",
```

```
    name    : "VIDEO_M3_DATA_MEM",
    base    : VIDEO_M3_DATA_ADDR,
    len     : VIDEO_M3_DATA_SIZE
}],
["VIDEO_M3_BSS_MEM", {
    comment : "VIDEO_M3_BSS_MEM",
    name    : "VIDEO_M3_BSS_MEM",
    base    : VIDEO_M3_BSS_ADDR,
    len     : VIDEO_M3_BSS_SIZE
}],
["VIDEO_M3_BSS_MAPPED_MEM", {
    comment : "VIDEO_M3_BSS_MAPPED_MEM",
    name    : "VIDEO_M3_BSS_MAPPED_MEM",
    base    : VIDEO_M3_BSS_MAPPED_ADDR,
    len     : VIDEO_M3_BSS_SIZE
}],
["DSS_M3_CODE_MEM", {
    comment : "DSS_M3_CODE_MEM",
    name    : "DSS_M3_CODE_MEM",
    base    : DSS_M3_CODE_ADDR,
    len     : DSS_M3_CODE_SIZE
}],
["DDR3_M3", {
    comment : "DDR3_M3",
    name    : "DDR3_M3",
    base    : DSS_M3_DATA_ADDR,
    len     : DSS_M3_DATA_SIZE
}],
["DSS_M3_BSS_MEM", {
    comment : "DSS_M3_BSS_MEM",
    name    : "DSS_M3_BSS_MEM",
    base    : DSS_M3_BSS_ADDR,
    len     : DSS_M3_BSS_SIZE
}],
["DSS_M3_BSS_MAPPED_MEM", {
    comment : "DSS_M3_BSS_MAPPED_MEM",
    name    : "DSS_M3_BSS_MAPPED_MEM",
    base    : DSS_M3_BSS_MAPPED_ADDR,
    len     : DSS_M3_BSS_SIZE
}],
["DSP_CODE_MEM", {
    comment : "DSP_CODE_MEM",
    name    : "DSP_CODE_MEM",
    base    : DSP_CODE_ADDR,
    len     : DSP_CODE_SIZE
}],
["DSP_DATA_MEM", {
    comment : "DSP_DATA_MEM",
    name    : "DSP_DATA_MEM",
    base    : DSP_DATA_ADDR,
    len     : DSP_DATA_SIZE
}],
["TILER_MEM", {
    comment : "TILER_MEM",
    name    : "TILER_MEM",
    base    : TILER_ADDR,
    len     : TILER_SIZE
}],
["SR2_FRAME_BUFFER_MEM", {
    comment : "SR2_FRAME_BUFFER_MEM",
```

```
    name    : "SR2_FRAME_BUFFER_MEM",
    base    : SR2_FRAME_BUFFER_ADDR,
    len     : SR2_FRAME_BUFFER_SIZE
}],
["SR0", {
    comment : "SR0",
    name    : "SR0",
    base    : SR0_ADDR,
    len     : SR0_SIZE
}],
["VIDEO_M3_EXCEPTION_CTX", {
    comment : "VIDEO_M3_EXCEPTION_CTX",
    name    : "VIDEO_M3_EXCEPTION_CTX",
    base    : VIDEO_M3_EXCEPTION_CTX_ADDR,
    len     : VIDEO_M3_EXCEPTION_CTX_SIZE
}],
["VPSS_M3_EXCEPTION_CTX", {
    comment : "VPSS_M3_EXCEPTION_CTX",
    name    : "VPSS_M3_EXCEPTION_CTX",
    base    : VPSS_M3_EXCEPTION_CTX_ADDR,
    len     : VPSS_M3_EXCEPTION_CTX_SIZE
}],
["HDVPSS_DESC_MEM", {
    comment : "HDVPSS_DESC_MEM",
    name    : "HDVPSS_DESC_MEM",
    base    : HDVPSS_DESC_ADDR,
    len     : HDVPSS_DESC_SIZE
}],
["HDVPSS_SHARED_MEM", {
    comment : "HDVPSS_SHARED_MEM",
    name    : "HDVPSS_SHARED_MEM",
    base    : HDVPSS_SHARED_ADDR,
    len     : HDVPSS_SHARED_SIZE
}],
["HOST_VPSS_NOTIFYMEM", {
    comment : "HOST_VPSS_NOTIFYMEM",
    name    : "HOST_VPSS_NOTIFYMEM",
    base    : NOTIFY_SHARED_ADDR,
    len     : NOTIFY_SHARED_SIZE
}],
["REMOTE_DEBUG_MEM", {
    comment : "REMOTE_DEBUG_MEM",
    name    : "REMOTE_DEBUG_MEM",
    base    : REMOTE_DEBUG_ADDR,
    len     : REMOTE_DEBUG_SIZE
}],
["L2_ROM", {
    comment: "L2_ROM",
    name: "L2_ROM",
    base: 0x00000000,
    len:  0x00004000
}],
["OCMC1_RAM", {
    comment: "OCMC1_RAM",
    name: "OCMC1_RAM",
    base: OCMC1_ADDR,
    len:  OCMC_SIZE
}],
["OCMC1_RAM_MAPPED", {
    comment: "OCMC1_RAM",
```

```
            name: "OCMC1_RAM_MAPPED",
            base: OCMC1_RUN_ADDR,
            len:  OCMC_SIZE
        }],
    ]
};

Build.platformTable["ti.platforms.evmTI816X:core0"] =
{
    externalMemoryMap:
    [
        ["DDR3_RAM", {
            comment: "DDR3_RAM",
            name: "DDR3_RAM",
            base: DDR3_ADDR,
            len:  DDR3_SIZE
        }],
        ["OCMC0_RAM", {
            comment: "OCMC0_RAM",
            name: "OCMC0_RAM",
            base: OCMC0_ADDR,
            len:  OCMC_SIZE
        }],
        ["VIDEO_M3_BSS_MAPPED_MEM", {
            comment : "VIDEO_M3_BSS_MAPPED_MEM",
            name    : "VIDEO_M3_BSS_MAPPED_MEM",
            base    : VIDEO_M3_BSS_MAPPED_ADDR,
            len     : VIDEO_M3_BSS_SIZE
        }],
        ["DSS_M3_BSS_MAPPED_MEM", {
            comment : "DSS_M3_BSS_MAPPED_MEM",
            name    : "DSS_M3_BSS_MAPPED_MEM",
            base    : DSS_M3_BSS_MAPPED_ADDR,
            len     : DSS_M3_BSS_SIZE
        }],
        ["L2_SRAM", {
            comment: "L2_SRAM",
            name: "L2_SRAM",
            base: L2_SRAM_ADDR,
            len:  L2_SRAM_SIZE
        }],
        ["L2_SRAM_RUN", {
            comment: "L2_SRAM_RUN",
            name: "L2_SRAM_RUN",
            base: L2_SRAM_RUN_ADDR,
            len:  L2_SRAM_SIZE
        }],
    ],
    customMemoryMap:
    [
        ["LINUX_MEM", {
            comment : "LINUX_MEM",
            name    : "LINUX_MEM",
            base    : LINUX_ADDR,
            len     : LINUX_SIZE
        }],
        ["SR1", {
            comment : "SR1",
            name    : "SR1",
            base    : SR1_ADDR,
```

```
    len     : SR1_SIZE
}],
["SR3_INTRADUCATI_IPC", {
    comment : "SR3_INTRADUCATI_IPC",
    name    : "SR3_INTRADUCATI_IPC",
    base    : SR3_INTRADUCATI_IPC_ADDR,
    len     : SR3_INTRADUCATI_IPC_SIZE
}],
["VIDEO_M3_CODE_MEM", {
    comment : "VIDEO_M3_CODE_MEM",
    name    : "VIDEO_M3_CODE_MEM",
    base    : VIDEO_M3_CODE_ADDR,
    len     : VIDEO_M3_CODE_SIZE
}],
["DDR_M3", {
    comment : "DDR3_M3",
    name    : "DDR3_M3",
    base    : VIDEO_M3_DATA_ADDR,
    len     : VIDEO_M3_DATA_SIZE
}],
["VIDEO_M3_BSS_MEM", {
    comment : "VIDEO_M3_BSS_MEM",
    name    : "VIDEO_M3_BSS_MEM",
    base    : VIDEO_M3_BSS_ADDR,
    len     : VIDEO_M3_BSS_SIZE
}],
["VIDEO_M3_BSS_MAPPED_MEM", {
    comment : "VIDEO_M3_BSS_MAPPED_MEM",
    name    : "VIDEO_M3_BSS_MAPPED_MEM",
    base    : VIDEO_M3_BSS_MAPPED_ADDR,
    len     : VIDEO_M3_BSS_SIZE
}],
["DSS_M3_CODE_MEM", {
    comment : "DSS_M3_CODE_MEM",
    name    : "DSS_M3_CODE_MEM",
    base    : DSS_M3_CODE_ADDR,
    len     : DSS_M3_CODE_SIZE
}],
["DSS_M3_BSS_MEM", {
    comment : "DSS_M3_BSS_MEM",
    name    : "DSS_M3_BSS_MEM",
    base    : DSS_M3_BSS_ADDR,
    len     : DSS_M3_BSS_SIZE
}],
["DSS_M3_BSS_MAPPED_MEM", {
    comment : "DSS_M3_BSS_MAPPED_MEM",
    name    : "DSS_M3_BSS_MAPPED_MEM",
    base    : DSS_M3_BSS_MAPPED_ADDR,
    len     : DSS_M3_BSS_SIZE
}],
["DSS_M3_DATA_MEM", {
    comment : "DSS_M3_DATA_MEM",
    name    : "DSS_M3_DATA_MEM",
    base    : DSS_M3_DATA_ADDR,
    len     : DSS_M3_DATA_SIZE
}],
["DSP_CODE_MEM", {
    comment : "DSP_CODE_MEM",
    name    : "DSP_CODE_MEM",
    base    : DSP_CODE_ADDR,
```

```
    len     : DSP_CODE_SIZE
}],
["DSP_DATA_MEM", {
    comment : "DSP_DATA_MEM",
    name    : "DSP_DATA_MEM",
    base    : DSP_DATA_ADDR,
    len     : DSP_DATA_SIZE
}],
["TILER_MEM", {
    comment : "TILER_MEM",
    name    : "TILER_MEM",
    base    : TILER_ADDR,
    len     : TILER_SIZE
}],
["SR2_FRAME_BUFFER_MEM", {
    comment : "SR2_FRAME_BUFFER_MEM",
    name    : "SR2_FRAME_BUFFER_MEM",
    base    : SR2_FRAME_BUFFER_ADDR,
    len     : SR2_FRAME_BUFFER_SIZE
}],
["SR0", {
    comment : "SR0",
    name    : "SR0",
    base    : SR0_ADDR,
    len     : SR0_SIZE
}],
["VIDEO_M3_EXCEPTION_CTX", {
    comment : "VIDEO_M3_EXCEPTION_CTX",
    name    : "VIDEO_M3_EXCEPTION_CTX",
    base    : VIDEO_M3_EXCEPTION_CTX_ADDR,
    len     : VIDEO_M3_EXCEPTION_CTX_SIZE
}],
["VPSS_M3_EXCEPTION_CTX", {
    comment : "VPSS_M3_EXCEPTION_CTX",
    name    : "VPSS_M3_EXCEPTION_CTX",
    base    : VPSS_M3_EXCEPTION_CTX_ADDR,
    len     : VPSS_M3_EXCEPTION_CTX_SIZE
}],
["HDVPSS_DESC_MEM", {
    comment : "HDVPSS_DESC_MEM",
    name    : "HDVPSS_DESC_MEM",
    base    : HDVPSS_DESC_ADDR,
    len     : HDVPSS_DESC_SIZE
}],
["HDVPSS_SHARED_MEM", {
    comment : "HDVPSS_SHARED_MEM",
    name    : "HDVPSS_SHARED_MEM",
    base    : HDVPSS_SHARED_ADDR,
    len     : HDVPSS_SHARED_SIZE
}],
["HOST_VPSS_NOTIFYMEM", {
    comment : "HOST_VPSS_NOTIFYMEM",
    name    : "HOST_VPSS_NOTIFYMEM",
    base    : NOTIFY_SHARED_ADDR,
    len     : NOTIFY_SHARED_SIZE
}],
["REMOTE_DEBUG_MEM", {
    comment : "REMOTE_DEBUG_MEM",
    name    : "REMOTE_DEBUG_MEM",
    base    : REMOTE_DEBUG_ADDR,
```

```
            len     : REMOTE_DEBUG_SIZE
        }],
        ["L2_SRAM", {
            comment: "L2_SRAM",
            name: "L2_SRAM",
            base: L2_SRAM_ADDR,
            len:  L2_SRAM_SIZE
        }],
        ["L2_SRAM_RUN", {
            comment: "L2_SRAM_RUN",
            name: "L2_SRAM_RUN",
            base: L2_SRAM_RUN_ADDR,
            len:  L2_SRAM_SIZE
        }],
        ["L2_ROM", {
            comment: "L2_ROM",
            name: "L2_ROM",
            base: 0x00000000,
            len:  0x00004000
        }],
        ["OCMC0_RAM", {
            comment: "OCMC0_RAM",
            name: "OCMC0_RAM",
            base: OCMC0_ADDR,
            len:  OCMC_SIZE
        }],
        ["OCMC0_RAM_MAPPED", {
            comment: "OCMC0_RAM",
            name: "OCMC0_RAM_MAPPED",
            base: OCMC0_RUN_ADDR,
            len:  OCMC_SIZE
        }],
    ]
};

Build.platformTable["ti.platforms.evmTI816X:plat"] =
{
    externalMemoryMap:
    [
        ["DDR3_RAM", {
            comment: "DDR3_RAM",
            name: "DDR3_RAM",
            base: DDR3_ADDR,
            len:  DDR3_SIZE
        }],
        ["OCMC0_RAM", {
            comment: "OCMC0_RAM",
            name: "OCMC0_RAM",
            base: OCMC0_ADDR,
            len:  OCMC_SIZE
        }],
        ["OCMC1_RAM", {
            comment: "OCMC1_RAM",
            name: "OCMC1_RAM",
            base: OCMC1_ADDR,
            len:  OCMC_SIZE
        }],
    ],
    customMemoryMap:
    [
```

```
["LINUX_MEM", {
    comment : "LINUX_MEM",
    name    : "LINUX_MEM",
    base    : LINUX_ADDR,
    len     : LINUX_SIZE
}],
["SR1", {
    comment : "SR1",
    name    : "SR1",
    base    : SR1_ADDR,
    len     : SR1_SIZE
}],
["SR3_INTRADUCATI_IPC", {
    comment : "SR3_INTRADUCATI_IPC",
    name    : "SR3_INTRADUCATI_IPC",
    base    : SR3_INTRADUCATI_IPC_ADDR,
    len     : SR3_INTRADUCATI_IPC_SIZE
}],
["VIDEO_M3_CODE_MEM", {
    comment : "VIDEO_M3_CODE_MEM",
    name    : "VIDEO_M3_CODE_MEM",
    base    : VIDEO_M3_CODE_ADDR,
    len     : VIDEO_M3_CODE_SIZE
}],
["VIDEO_M3_DATA_MEM", {
    comment : "VIDEO_M3_DATA_MEM",
    name    : "VIDEO_M3_DATA_MEM",
    base    : VIDEO_M3_DATA_ADDR,
    len     : VIDEO_M3_DATA_SIZE
}],
["VIDEO_M3_BSS_MEM", {
    comment : "VIDEO_M3_BSS_MEM",
    name    : "VIDEO_M3_BSS_MEM",
    base    : VIDEO_M3_BSS_ADDR,
    len     : VIDEO_M3_BSS_SIZE
}],
["DSS_M3_CODE_MEM", {
    comment : "DSS_M3_CODE_MEM",
    name    : "DSS_M3_CODE_MEM",
    base    : DSS_M3_CODE_ADDR,
    len     : DSS_M3_CODE_SIZE
}],
["DSS_M3_DATA_MEM", {
    comment : "DSS_M3_DATA_MEM",
    name    : "DSS_M3_DATA_MEM",
    base    : DSS_M3_DATA_ADDR,
    len     : DSS_M3_DATA_SIZE
}],
["DSS_M3_BSS_MEM", {
    comment : "DSS_M3_BSS_MEM",
    name    : "DSS_M3_BSS_MEM",
    base    : DSS_M3_BSS_ADDR,
    len     : DSS_M3_BSS_SIZE
}],
["DSP_CODE_MEM", {
    comment : "DSP_CODE_MEM",
    name    : "DSP_CODE_MEM",
    base    : DSP_CODE_ADDR,
    len     : DSP_CODE_SIZE
}],
```

```
["DSP_DATA_MEM", {
    comment : "DDR3_DSP",
    name    : "DDR3_DSP",
    base    : DSP_DATA_ADDR,
    len     : DSP_DATA_SIZE
}],
["TILER_MEM", {
    comment : "TILER_MEM",
    name    : "TILER_MEM",
    base    : TILER_ADDR,
    len     : TILER_SIZE
}],
["SR2_FRAME_BUFFER_MEM", {
    comment : "SR2_FRAME_BUFFER_MEM",
    name    : "SR2_FRAME_BUFFER_MEM",
    base    : SR2_FRAME_BUFFER_ADDR,
    len     : SR2_FRAME_BUFFER_SIZE
}],
["SR0", {
    comment : "SR0",
    name    : "SR0",
    base    : SR0_ADDR,
    len     : SR0_SIZE
}],
["VIDEO_M3_EXCEPTION_CTX", {
    comment : "VIDEO_M3_EXCEPTION_CTX",
    name    : "VIDEO_M3_EXCEPTION_CTX",
    base    : VIDEO_M3_EXCEPTION_CTX_ADDR,
    len     : VIDEO_M3_EXCEPTION_CTX_SIZE
}],
["VPSS_M3_EXCEPTION_CTX", {
    comment : "VPSS_M3_EXCEPTION_CTX",
    name    : "VPSS_M3_EXCEPTION_CTX",
    base    : VPSS_M3_EXCEPTION_CTX_ADDR,
    len     : VPSS_M3_EXCEPTION_CTX_SIZE
}],
["HDVPSS_DESC_MEM", {
    comment : "HDVPSS_DESC_MEM",
    name    : "HDVPSS_DESC_MEM",
    base    : HDVPSS_DESC_ADDR,
    len     : HDVPSS_DESC_SIZE
}],
["HDVPSS_SHARED_MEM", {
    comment : "HDVPSS_SHARED_MEM",
    name    : "HDVPSS_SHARED_MEM",
    base    : HDVPSS_SHARED_ADDR,
    len     : HDVPSS_SHARED_SIZE
}],
["HOST_VPSS_NOTIFYMEM", {
    comment : "HOST_VPSS_NOTIFYMEM",
    name    : "HOST_VPSS_NOTIFYMEM",
    base    : NOTIFY_SHARED_ADDR,
    len     : NOTIFY_SHARED_SIZE
}],
["REMOTE_DEBUG_MEM", {
    comment : "REMOTE_DEBUG_MEM",
    name    : "REMOTE_DEBUG_MEM",
    base    : REMOTE_DEBUG_ADDR,
    len     : REMOTE_DEBUG_SIZE
}],
```

```
        ["OCMC0_RAM", {
            comment: "OCMC0_RAM",
            name: "OCMC0_RAM",
            base: OCMC0_ADDR,
            len:  OCMC_SIZE
        }],
        ["OCMC1_RAM", {
            comment: "OCMC1_RAM",
            name: "OCMC1_RAM",
            base: OCMC1_ADDR,
            len:  OCMC_SIZE
        }],
        ["DSP_L2_RAM", {
            comment: "DSP_L2_RAM",
            name: "DSP_L2_RAM",
            base: 0x10800000,
            len:  0x00020000
        }],
    ],
    l1PMode: "32k",
    l1DMode: "32k",
    l2Mode:  "128k"
};

var addrFileGenerated = false;
if (addrFileGenerated == false)
{
    xdc.loadCapsule("genaddrinfo.xs").GenAddrFile();
    addrFileGenerated = true;
}
```