o SW reference decoder is the standard reference implementation downloaded from net from the corresponding organization for HEVC,AVC,VP9 and AV1.

o HW reference decoder is the intel media engine decoder on the platform.

**HEVC:**

**Installation steps**:

$ git clone https://vcgit.hhi.fraunhofer.de/jct-vc/HM.git hm\_1

$ mkdir build && cd build

$ cmake ..

$ make

$ make install

**Commands:**

**HEVC Software decoder :**

./TAppDecoderStatic -b Allegro\_HEVC\_Main\_HT50\_BANDWIDTH\_00-CTB32\_3840x2160@30Hz\_r5.8\_4K.bin -o Allegro\_HEVC\_Main\_HT50\_BANDWIDTH\_00-CTB32\_3840x2160@30Hz\_r5.8\_4K\_SWRef.yuv

/home/testuser/Documents/Media\_VDBox\_PVC/HWDecoders/developer\_pkg/lin\_x64/bin/./mfx\_player -i:hevc

**HEVC HW decoder:**

/home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/hevc\_4k\_res/hevc\_swr/Allegro\_HEVC\_Main\_HT50\_BANDWIDTH\_00-CTB32\_3840x2160@30Hz\_r5.8\_4K.bin -no\_render -hw -priority 1 -perf\_opt -d3d -o /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/hevc\_4k\_res/hevc\_swr/Allegro\_HEVC\_Main\_HT50\_BANDWIDTH\_00-CTB32\_3840x2160@30Hz\_r5.8\_4K\_HW.yuv

**HEVC compare:**

./comp\_hevc\_yuv Allegro\_HEVC\_Main\_HT50\_BANDWIDTH\_00-CTB32\_3840x2160@30Hz\_r5.8\_4K\_HW.yuv Allegro\_HEVC\_Main\_HT50\_BANDWIDTH\_00-CTB32\_3840x2160@30Hz\_r5.8\_4K\_SWRef.yuv >> Log\_hevc.txt

**AVC:**

**Installation steps:**

download from this path and extract the JM

git clone http://iphome.hhi.de/suehring/tml/download/ JM

run this Commands

-->$ . unixprep.sh

-->$ make

**Commands:**

**AVC software decoder:**

/home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AVC\_TEST/AVC\_SW\_Decorder/jm19.0/JM/bin/./ldecod.exe -p InputFile=/home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AVC\_TEST/3\_rd\_test/avc\_swref/V30355\_A\_D-Traffic\_AVC\_MP\_playback\_625SD.264 -p OutputFile=/home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AVC\_TEST/3\_rd\_test/avc\_swref/V30355\_A\_D-Traffic\_AVC\_MP\_playback\_625SD\_SWRef.yuv >> AVCref.txt

**AVC HW decoder:**

/home/ubntu1804/Documents/Media\_VDBox\_PVC/HWDecoders/developer\_pkg/lin\_x64/bin/./mfx\_player -i:h264 /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AVC\_TEST/3\_rd\_test/avc\_hw/V30355\_A\_D-Traffic\_AVC\_MP\_playback\_625SD.264 -no\_render -hw -priority 1 -perf\_opt -d3d -o /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AVC\_TEST/3\_rd\_test/avc\_hw/V30355\_A\_D-Traffic\_AVC\_MP\_playback\_625SD\_HW.yuv >> mfx\_AVC\_cmd\_line.txt

**AVC compare:**

./comp\_avc\_yuv V30355\_A\_D-Traffic\_AVC\_MP\_playback\_625SD\_HW.yuv V30355\_A\_D-Traffic\_AVC\_MP\_playback\_625SD\_SWRef.yuv >> Log\_avc.txt

**VP9:**

**Installation steps:**

More information :http://www.linuxfromscratch.org/blfs/view/svn/multimedia/libvpx.html

git clone https://chromium.googlesource.com/webm/libvpx/

run this entire command:

sed -i 's/cp -p/cp/' build/make/Makefile &&

mkdir libvpx-build &&

cd libvpx-build &&

../configure --prefix=/usr \

             --enable-shared \

             --disable-static &&

make

**Commands :**

**vp9 software decoder**:

/home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/vp9\_tests/libvpx/libvpx-build/./vpxdec --i420 -o /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/vp9\_tests/vp9\_swf/blue\_sky\_1080p\_5s\_30fps\_SWRef.yuv /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/vp9\_tests/vp9\_hw/blue\_sky\_1080p\_5s\_30fps.ivf

**vp9 HW decoder**:

~/Media/msdk\_test/lin\_x64/bin/./mfx\_player --input /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/vp9\_tests/vp9\_hw/blue\_sky\_1080p\_5s\_30fps.ivf -no\_render -hw -priority 1 -perf\_opt -d3d -o /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/vp9\_tests/vp9\_hw/blue\_sky\_1080p\_5s\_30fps\_HW.yuv >> mfx\_VP9\_cmd\_line.txt

**vp9 compare**:

./comp\_vp9\_yuv /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/vp9\_tests/compare/blue\_sky\_1080p\_5s\_30fps\_HW.yuv /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/vp9\_tests/compare/blue\_sky\_1080p\_5s\_30fps\_SWRef.yuv >> Log\_vp9.txt

**AV1:**

**Installation steps**:

for more information:

https://aomedia.googlesource.com/aom/

sudo apt-get update -y

sudo apt-get install -y yasm

git clone https://github.com/emscripten-core/emsdk.git

cd emsdk

# Fetch the latest version of the emsdk (not needed the first time you clone)

git pull

# Download and install the latest SDK tools.

./emsdk install latest

# Make the "latest" SDK "active" for the current user. (writes .emscripten file)

./emsdk activate latest

# Activate PATH and other environment variables in the current terminal

source ./emsdk\_env.sh

$git clone https://aomedia.googlesource.com/aom

$cd aom

$ cmake ../../aom

$ make

**Commands:**

**AV1 Software decoder**:

./aomdec --output-bit-depth=8 --rawvideo -o /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AV1\_TEST/AV1\_SWRef/test/720p50\_mobcal\_5s\_30fps\_av1\_SWRef.yuv /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AV1\_TEST/AV1\_SWRef/test/720p50\_mobcal\_5s\_30fps\_av1.ivf

**AV1 HW decoder**

./mfx\_player -i:av1 /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AV1\_TEST/AV1\_SWRef/test/720p50\_mobcal\_5s\_30fps\_av1.ivf -hw -no\_render -priority 1 -perf\_opt -d3d -o /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AV1\_TEST/AV1\_SWRef/test/720p50\_mobcal\_5s\_30fps\_av1\_HW.yuv >> mfx\_AV1\_cmd\_line.txt

**AV1 compare** :

./comp\_av1\_yuv /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AV1\_TEST/AV1\_SWRef/test/720p50\_mobcal\_5s\_30fps\_av1\_HW.yuv /home/ubntu1804/Documents/Media\_VDBox\_PVC/NEW\_TESTS\_EXEC/AV1\_TEST/AV1\_SWRef/test/720p50\_mobcal\_5s\_30fps\_av1\_SwRef.yuv >> Log\_av1.txt