In [11]:

```
from pynq.overlays.base import BaseOverlay
base = BaseOverlay("base.bit")
from pynq.lib.video import *
import cv2
import numpy as np
import time

CONTRAST_THRESHOLD = 80
HISTOGRAM_SIZE = 256
mini = 0
maxi = 255
diff = 255
histogram = np.zeros(HISTOGRAM_SIZE, dtype = np.uint)
```

In [12]:

```
hdmi_in = base.video.hdmi_in
hdmi_out = base.video.hdmi_out
hdmi_in.configure(PIXEL_RGB)
hdmi_out.configure(hdmi_in.mode, PIXEL_RGB)
hdmi_in.start()
hdmi_out.start()
```

Out[12]:

```
<contextlib._GeneratorContextManager at 0x23fcc330>
```

In [13]:

```
start = time.time()
height = hdmi_in.mode.height
width = hdmi_in.mode.width
gray_frame = np.ndarray(shape=(height,
                              width), dtype=np.uint8)
numframes = 10
for _ in range(numframes):
    capture = hdmi_in.readframe()
    for i in range(0,height):
        for j in range(0,width):
            gray_frame[i,j] = np.uint8(0.3*capture[i,j,2] + 0.59*capture[i,j,1]
+ 0.11*capture[i,j,0])
            histogram[gray_frame[i,j]] = histogram[gray_frame[i,j]] + 1
            if(gray_frame[i,j] < mini):
                gray_frame[i,j] = 0
            elif(gray_frame[i,j] > maxi):
                gray_frame[i,j] = 255
            else: #if((gray_frame[i,j] >= mini) and (gray_frame[i,j] <= maxi)):
                gray_frame[i,j] = np.uint8((maxi*(gray_frame[i,j]-mini)/diff))

    x = 0
    while((x < HISTOGRAM_SIZE) and (histogram[x] < CONTRAST_THRESHOLD)):
        x = x + 1
    mini = x
    x = HISTOGRAM_SIZE - 1
    while((x > mini) and (histogram[x] < CONTRAST_THRESHOLD)):
        x = x - 1
    maxi = x
    diff = maxi - mini
    outframe = hdmi_out.newframe()
    cv2.cvtColor(gray_frame, cv2.COLOR_GRAY2RGB,dst=outframe)
    hdmi_out.writeframe(outframe)
    capture.freebuffer()

end = time.time()
print("Frames per second:  " + str(numframes / (end - start)))
```

```
Frames per second:  0.0023909867477575255
```

In [10]:

```
hdmi_out.stop()
hdmi_in.stop()
del hdmi_in, hdmi_out
```

In [ ]:

```
```