In [9]:
```
from pynq.overlays.base import BaseOverlay
base = BaseOverlay("base.bit")
from pynq.lib.video import *
import cv2
import numpy as np
import time
```

In [10]:
```
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[10]:
```
<contextlib._GeneratorContextManager at 0x2f4c1370>
```

In [11]:
```
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()

    cv2.cvtColor(capture, cv2.COLOR_RGB2GRAY,dst=gray_frame)
    capture.freebuffer()

    outframe = hdmi_out.newframe()
    cv2.cvtColor(gray_frame, cv2.COLOR_GRAY2RGB,dst=outframe)
    hdmi_out.writeframe(outframe)

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

```
Frames per second:  4.673391599045378
```

In [12]:
```
hdmi_out.stop()
hdmi_in.stop()
del hdmi_in, hdmi_out
```

In [ ]:
```

```