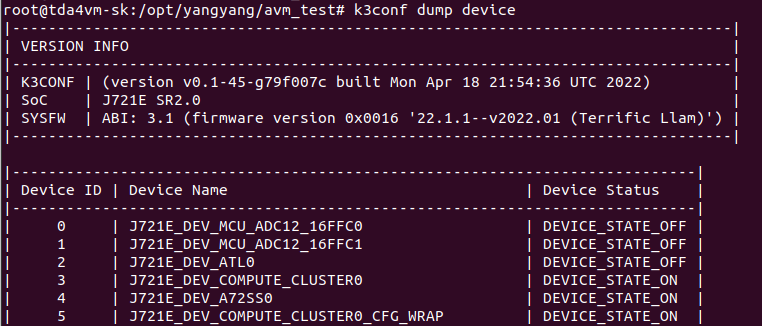
I use the following command to set the frequency:（DEV\_A72SS0\_ARM\_CLK\_CLK）：k3conf set clock <dev\_id> <clk\_id> <freq>

here：dev\_id=4, clk\_id=2



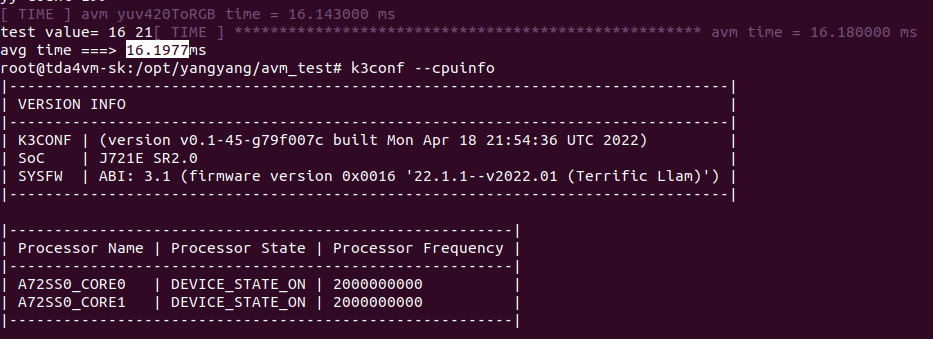


There is a demo program of YUV420 to RGB. The resolution of the input image is 1280\*960，It takes about 15ms to execute on TDA4，Is this a reasonable time？

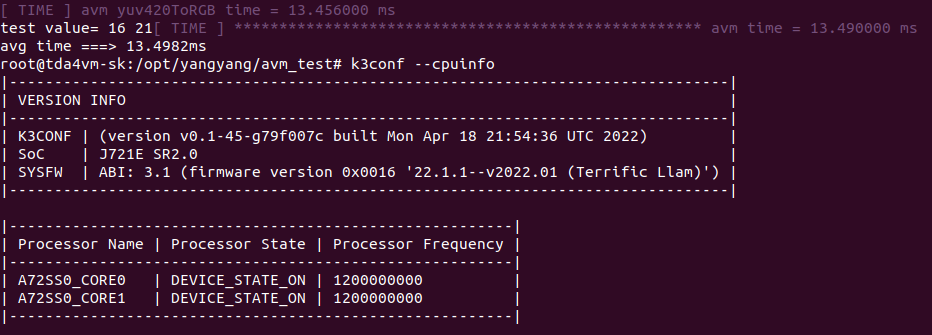
The performance of different frequencies are as follows:

|  |  |
| --- | --- |
| freqency（G Hz） | Avg Time（ms） |
| 2 | 16.1977 |
| 1.2 | 13.4982 |
| 0.8 | 13.5546 |
| 0.4 | 13.4973 |
| 0.1 | 13.4965 |
| 2.7 | 12.0583 |

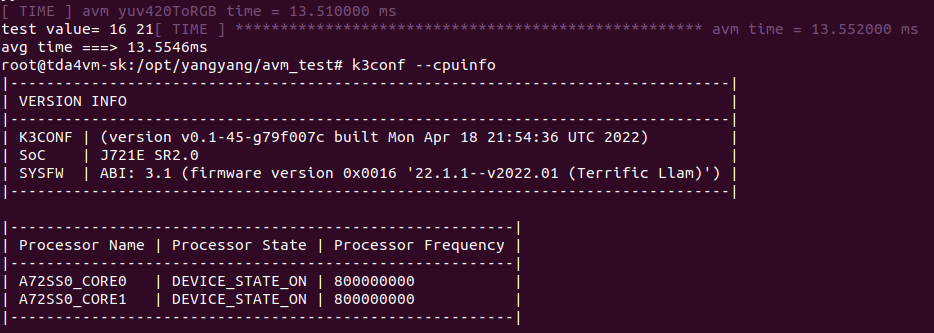
2GHz performance：



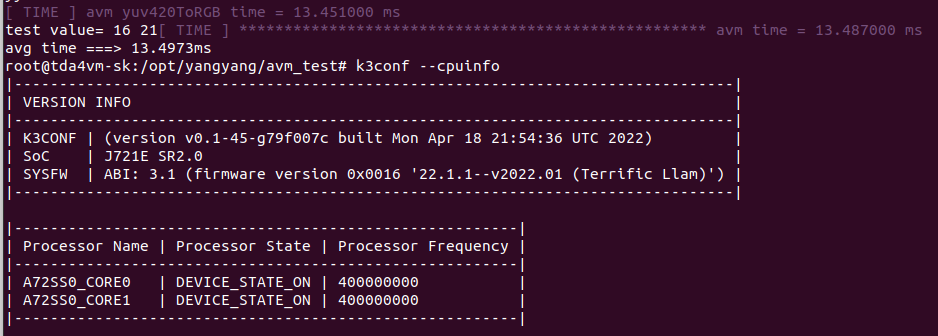
1.2GHz performance：



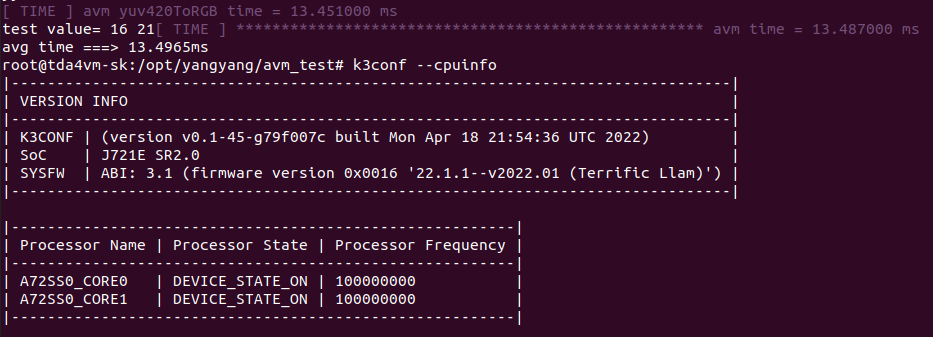
0.8GHz performance：



0.4GHz performance：



0.1GHz performance：



2.7GHz performance：



The source code of YUV420 to RGB is as follows：

void yuv420ToRGB(const uint8\_t\* pYuv, uint8\_t\* pRgb, const uint32\_t width, const uint32\_t height)

{

uint32\_t width\_min = width >> 1;

uint32\_t height\_min = height >> 1;

const uint8\_t\* Y = pYuv;

const uint8\_t\* U = Y + width \* height;

const uint8\_t\* V = U + width\_min \* height\_min;

timeval time\_1{}, time\_2{};

gettimeofday(&time\_1, nullptr);

for (uint32\_t j = 0; j < height; j++)

{

for (uint32\_t i = 0; i < width; i++)

{

uint8\_t y = Y[width \* j + i];

uint8\_t u = U[(j >> 1) \* width\_min + (i >> 1)];

uint8\_t v = V[(j >> 1) \* width\_min + (i >> 1)];

int16\_t r = y + 1.402 \* (v - 128);

int16\_t g = y - 0.34414 \* (u - 128) - 0.71414 \* (v - 128);

int16\_t b = y + 1.772 \* (u - 128);

pRgb[(j \* width + i ) \* 3 + 0] = b < 0 ? 0 : ((b > 255) ? 255 : b);

pRgb[(j \* width + i ) \* 3 + 1] = g < 0 ? 0 : ((g > 255) ? 255 : g);

pRgb[(j \* width + i ) \* 3 + 2] = r < 0 ? 0 : ((r > 255) ? 255 : r);

}

}

DELTA\_TIME(time\_1, time\_2, "avm yuv420ToRGB time = %f ms");

}