An Industry Standard Benchmark Consortium

ConsumerBench™ Version 1.1

Benchmark Name: RGB to YIQ Conversion

Highlights

- **Benchmarks performance for digital video processing.**
- **Explores multiply / accumulate capability.**
- **This benchmark has opportunities for Full-Fury benchmark optimizations, especially by SIMD and VLIW architectures.**

Application

RGB to YIQ conversion is used in the NTSC encoder where the RGB inputs from the camera are converted to a luminance (Y) and two chrominance information (I,Q). In the NTSC encoder, these I,Q signals are modulated by a subcarrier and added to the Y signal.

Historically, when color TVs appeared in the market, they had to coexist with the existing monochrome TVs and this was made possible with the NTSC signal structure. The chrominance signals are averaged out as a fine mesh of invisible signals in the monochrome TV sets.
YUV used in the European PAL standard and YCbCr used in the JPEG standard have different codings. All three standards share the same luminance signal Y but the chrominance calculations are different. The matrix calculation scheme used in the RGB to YIQ can be applied to these standards too.

In the actual products, this trivial calculation is usually performed in dedicated hardware, especially in digital video products. For cost saving and flexibility, this algorithm can be implemented in software if the CPU is powerful enough and where the digital image is a still picture.

Benchmark Description

This benchmark explores the capability of the CPU to perform a straightforward matrix multiply/accumulate calculation.
The R, G, B 8-bit pixel color image input is processed as follows:

```
Y = 0.299*R + 0.587*G + 0.114*B
I = 0.596*R – 0.275*G – 0.321*B
Q = 0.212*R – 0.523*G + 0.311*B
```

RGB values are in the range of [0:255]. The conversion coefficients are 16-bits. The multiply/accumulate results are shifted right by 16-bits. Before the shift, 1 is added to a bit location right to the LSB of the shifted result for rounding to the nearest integer.
The output is 8-bit data. Y is in the range of [0,255] and I,Q in the range of [-127, 127].

The input and output data size is 320-pixels in the horizontal direction and 240-pixels in the vertical direction. The 320x240 data for RGB and YIQ are stored sequentially as.
R[0], G[0], B[0], R[1], G[1], B[1],………R[76799], G[76799], B[76799]
Y[0], I[0], Q[0], Y[1], I[1], Q[1],………Y[76799], I[76799], Q[76799]
The pointers are just incremented by one to access R, G,B or Y, I, Q data is this order..

Analysis of Computing Resources

Out of the Box Benchmark: A ‘for loop’ calculates the conversion of a set of RGB inputs and YIQ outputs at a time. A set of R, G, B input data is read from the memory by incrementing a read pointer. A set of output Y, I, Q output data is written back to the memory by incrementing a write pointer. There is no complex 2-dimensonal access such as that in the high pass grey-scale filter benchmark.

The calculation is a straightforward multiplication and accumulation that a microprocessor with a single-cycle MAC unit will benefit from.

The code size is trivial and easily fits in to a small L1 Instruction Cache.

Full-Fury Benchmark: Because of the simple structure of the multiplication and accumulation, a VLIW or SIMD architecture with multiple of MAC units can be used to accelerate performance. A further optimization is the loading of multiple Bytes at a time. Software pipelining could be used to pass the loaded data efficiently to the MAC unit for calculation.