An Industry Standard Benchmark Consortium

ConsumerBench™ Version 1.1

Benchmark Name: RGB to CMYK Conversion

Highlights

- **Benchmarks digital image processing performance in printers and other digital imaging products.**
- **Explores basic arithmetic and minimum value detection capability.**
- **This benchmark provides opportunities for Full-Fury benchmark optimization. Conditional move and multi-Byte processing SIMD or VLIW architectures are effective for example.**

Application

RGB to CMYK conversion is widely used in color printers. RGB inputs from PC data is converted to CMYK color signals for printing..

Benchmark Description

This benchmark explores the target CPU capability for basic arithmetic and minimum value detection.

R, G, B 8-bit pixel color image input is fed to the following equation:

```
/* calculate complementary colors */
c = 255 – R;
m = 255 – G;
y = 255 – B;

/* find the black level k */
K = minimum (c,m,y)

/* correct complementary color lever based on k */
C = c – K
M = m – K
Y = y - K
```

RGB values are in the range of [0:255]..

CMYK values are in the range of [0:255]..

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 CMYK is stored sequentially as.

R[0], G[0], B[0], R[1], G[1], B[1],………………R[76799], G[76799], B[76799]

C[0], M[0], Y[0], K[0], C[1], M[1], Y[1],K[1]……C[76799], M[76799], Y[76799], K[76799]

The pointers are just incremented by one to access R, G, B or C, M, Y, K data is this order.

If the benchmark score is extrapolated for a larger image, the processing time will be almost linearly proportional to the pixel count. (e.g. For a 640 x 480 image, it will be x4 times.) The iteration/sec score will be the inverse e.g. for a 640 x 480 image, iteration/sec it will be x1/4. There is data dependency in the cycle counts for the minimum value K search, due to branch taken or not taken. If this operation is handled by conditional move, the cycle will constant.

Analysis of Computing

Out of the Box Benchmark: A ‘for loop’ calculates the conversion of a set of RGB inputs and CMYK outputs at a time. A set of R, G, B input data is read from the memory by incrementing a

Resources

read pointer. A set of output C, M, Y, K output data is written back to the memory by incrementing a write pointer. There is no complex 2-dimensonal access like the high pass grey-scale filter benchmark.

The complementary color calculation and correction are simple subtract calculations without any MAC operation.

The minimum value search has two branches for processing each pixel.

```
If (c<m) {
            K = (Byte)(c<y ? c:y);
        }
    else {
            K = (Byte)(m<y ? m:y);
        }
```

This can be a very expensive routine because of the branch penalty.

Full-Fury Benchmark: By using compare and conditional moves, the branch penalty can be avoided. VLIW and SIMD can process multiple Byte of data at a time. A SIMD architecture which can handle multiple of Byte data at a time, is especially suited to this benchmark e.g. A 4-way SIMD microprocessor can handle 4 x 8-bit data every cycle.

Special Notes

Regarding the memory architecture, the image data is used just once and there is no benefit from a big Data Cache, unless the microprocessor has a cache prefetch feature. A small Data Cache will work to fetch consecutive data and avoid external memory access overhead. The code size is trivial and easily fits in to a small L1 Instruction Cache.