Low Power VLSI CMOS Design
An Image Processing Chip for RGB to HSI Conversion

A.Th. Schwarzbacher[1,2] and J.B. Foley[2]

[1]Dublin Institute of Technology, Dept. Of Electronic and Communication Eng., Dublin, Ireland.
schwarzbacher@electronics.dit.ie
[2]Trinity College, Dept. of Microelectronics and Electronic Eng., Dublin, Ireland.
bfoley@ee.tcd.ie

Abstract: In the early 90's, the first portable computing applications became widely available. During the early days of portable computing lower performance than mains depended counterparts was accepted. Today users though, demand small lightweight portable systems with long battery operating times without any compromise in computing performance. To meet these customer demands a consequent low power approach starting at the first design decision has to be taken. This paper will present the implementation of a low-power real-time image processing circuit for the transformation of a camera signal into the human perception code.

Keywords: Low-Power CMOS Design, RGB to HSI Conversion.

1 Introduction

The aim of this paper is to present a system level approach for the low-power implementation of computationally intensive algorithms. For this purpose Kender's algorithm for faster computing of hue [1], as shown in Table 1, was chosen as a means to investigate general valid methods of reducing the power consumption at the first stages of the VLSI design cycle. The traditional method of reducing the supply voltage [2] was not applicable for this task as the design was to be mapped into an ASIC library. Here voltage scaling is only possible in a very limited range. It therefore became clear that the

Kender's Algorithm for
Faster Computation of HUE:

```
if ((R > B) and (G > B)
    $hue = \frac{\pi}{3} + \arctan\left(\frac{\sqrt{3}\times(R-G)}{R-B+G-B}\right)$
else if (G > R)
    $hue = \pi + \arctan\left(\frac{\sqrt{3}\times(B-G)}{B-R+G-R}\right)$
else if (B > G)
    $hue = \frac{5\times\pi}{3} + \arctan\left(\frac{\sqrt{3}\times(R-B)}{R-G+B-G}\right)$
else if (R > B)
    $hue = 0$
else
        'achromatic'
```

Saturation:

$$saturation = 1 - \frac{3\times\min(R,G,B)}{R+G+B}$$

Intensity:

$$intensity = \frac{R+G+B}{3}$$

Table 1: The HSI Algorithm using Kender's Algorithm for the Faster Computation of Hue

only possibility to minimise the power consumption in such a design was to reduce the active capacitance.

Following this, the RGB to HSI algorithm was investigated. To enable a detailed analysis, the design was split into the three main paths: one for computing hue, one for the saturation and one for the intensity. These paths were again divided into smaller blocks, each containing a typical set of implementation problems. The block diagram of the circuit is shown in Figure 1. These blocks could then be investigated separately for potential reductions in power consumption. In the hue path the main task was to implement the alternating function of the arctan using unsigned arithmetic. This was achieved through the use of sign detection in the first stage of the design. This resulted in a smaller logic in the remaining stages. The second task was the implementation of the arctan function itself. The standard implementation for all trigonometric functions is the CORDIC algorithm. However the CORDIC is a general purpose algorithm. Due to the function of the arctan it was possible to design a small LUT [3]. This already yields power savings by a factor of 20. The transformation of the LUT also pointed into the direction of a hardware optimised approximation algorithm. This algorithm could realise the arctan function by using only one adder and one constant shift operation. Here a comparison of the CORDIC algorithm showed a reduced power consumption by a factor of 25 [4]. Furthermore, the other features such as the maximum frequency of operation or area requirements could be significantly reduced. Therefore, it was shown that the main strength of the CORDIC algorithm is in the area of mathematical multiprocessors rather than single function implementations. An investigation of the control pipeline showed a significant power inefficiency. First, different coding styles were applied to the design. However, the result was unsatisfying, as theoretically superior codes produced a higher power consumption than standard approaches. A detailed analysis showed that these power reduced codes had in fact a higher power consumption in the clock network of the pipeline. Therefore, an alternative to the traditional shift register implementation was developed. Here it was possible to show in a general study that even for small designs, such as that needed for the control bus, power savings of up to 30% could be achieved. For larger shift registers this figure increases even further.

The first design decision of the saturation path was the reuse of terms already computed in the hue path. This resulted in smaller logic, the use of balanced structures and a reduction in pipeline stages. Four different implementations were then considered. All implementations have theoretical advantages. However, the direct implementation of the saturation path showed the best power performance. This showed the difference between the software-optimised algorithms and hardware-optimised ones. While a direct implementation was not the smallest possible mathematical function, the block diagram nevertheless showed that it will be the smallest in terms of functional blocks. Also the investigation of the direct implementation underlined that this function was superior in terms of power consumption.

The last path to be implemented was the intensity algorithm. As in the case of the saturation

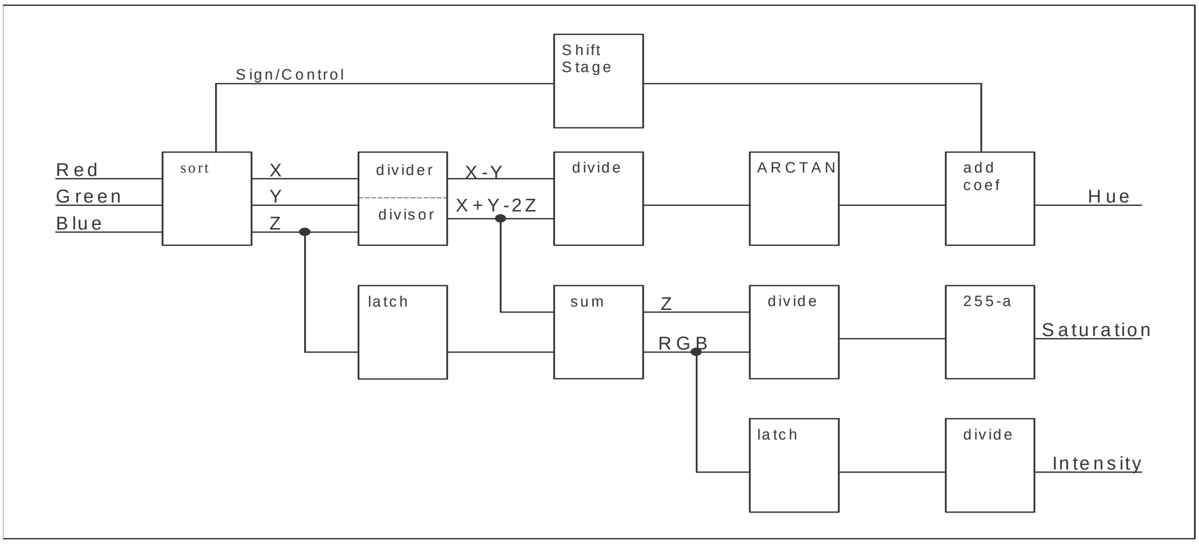

Figure 1: Block Diagram of the RGB to HSI Converter

algorithm, it was possible to reuse terms previously calculated for hue and intensity. Therefore, only one division by three had to be implemented. For such a constant divider various mathematical algorithms were developed. Although, so far there had been no investigation of their power consumption. The first finding was that several of these divider algorithms could be implemented with both alternating and non-alternating signs. As shown in [2], alternating implementations consume more power. Therefore, only the non alternating versions were investigated. Furthermore, two of the presented algorithms resulted in the same circuitry. Therefore, six different versions were implemented. An overall investigation showed that the algorithm proposed by Petry [6] gave the best power to area-speed performance. It was therefore used in this design. The accuracy of the intensity was also investigated. Here it was possible to replace the least significant bit by a constant ONE. This resulted in smaller logic and smaller power consumption while improving the accuracy by 33%.

2 Circuit Features

This section will present the performance of the implementation of the image processing algorithm. Table 2 summarises the features of the algorithm.

Features of the RGB to HSI Converter		
Technology	ES2 07μm (industrial)	
Supply Voltage	5V (±0.5V)	
Input Signals	Red, Green, Blue 8-bit unsigned Clock	
Output Signals	Hue, Saturation, Intensity 8-bit unsigned	
Throughput	30Mpixles / cycle	
Operating Frequency	30MHz	
	Low-Power Implementation	Direct Implementation
Area	4.19mm^2	3.7mm^2
Number of Pipeline Stages	5	4
Output Signal Deviation	Between -2 and +2 bit (0.78%)	Between -1 and +1 bit (0.39%)
Active Capacitance	187.9Ff	298.9fF
Maximum Settling Time	18ns	36ns
Maximum Throughput	50Mpixles / cycle	27.7Mpixles / cycle
Average Dynamic Power Consumption	140mW (@30MHz)	202mW (@30MHz)

Table 2: Performance of the RGB to HSI Converter

As presented in Table 2, the design was developed to convert images of a resolution of 1200 by 1200 pixels. After optimising the power consumption a peak resolution of up to 1600 by 1200 pixels at 25 frames per second was achieved. However, at this resolution the power consumption will rise by 39%, to 226mW, compared to the 1200 by 1200 pixel resolution.

Figure 2 shows the detailed breakdown of the individual components of the active capacitance in terms of RGB to Hue, Saturation and Intensity. Furthermore, it can be seen that the interconnect of the individual stages consumes the most power with 35%. If the intensity and hue subsystems are investigated further, it can be seen that the large divider structures present in both paths have the highest proportion of active capacitance, followed directly by the comparator in the hue path. To minimise the power consumption of these blocks, particular interest should be paid to the physical layout of these stages and the block interconnect.

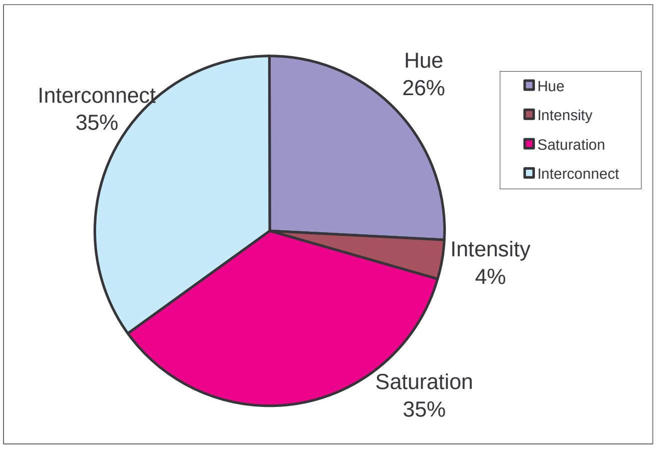

Figure 2: Breakdown of the Active Capacitance of the RGB to HSI Converter

In Table 2 the performance of the low power design is compared with a direct implementation of the RGB to HSI algorithm. This direct implementation of the algorithm uses the native VHDL operations to implement the various functions. Furthermore, the circuit was synthesised using design constraints to meet the timing requirements and not designed to meet constraints such as are required for the low-power design. As can be seen, the power consumption of the direct implementation is 37% higher than that of the power optimised implementation. Additionally, the maximum throughput, and therefore the maximum computational image resolution, was increased in the low-power version by a factor of 1.8. This increased performance is due to the balancing of the paths. In the low-power implementation paths which did not meet the timing requirements were shortened and hence the overall delay was decreased. In a voltage scalable circuit, this could be used for even further power reductions. In this case it would be possible to reduce the voltage down to approximately 3.5V in a full custom design which would result in a further reduction in power by half. The number of pipeline stages of the power optimised design was increased by one. This had the effect of balancing the paths and reducing the glitching in the divider structures. This caused a rise in latency by 18ns. However, if the latency of the power optimised design of 90ns is compared to that of the direct implementation, it can be seen that the additional pipeline stage did in fact reduce the latency by 54ns. The only feature of the low-power implementation which has not improved is the area. This result was anticipated. As has been shown in [2], the factor most often traded for a reduced power consumption is the area. The increase of 13% is very reasonable if it is compared to the reduced power consumption of 37% and increased performance of nearly 93%. While this direct implementation of the circuit does not perform as well as the optimised design, it has the advantage of a much faster design development cycle. Therefore, if fast time to market is of the uppermost importance to the designer, not all power saving features should be implemented. The use of gated clocks and the replacement of trigonometric functions by approximation algorithm is one method of effectively reducing the power consumption without the need to spend large amounts of time investigating the effects of alternative implementations.

After investigating the structural features, the image conversion properties are presented in Figure 3. On the right side the original picture is shown. In the middle the picture of the transformed algorithm is presented. Because there is no visible difference a subtraction picture is also included on the right hand side of Figure 3. Here small differences become visible. A theoretical, as well as a statistical investigation using a wide variety of images has shown that the maximum errors are limited to plus-minus two bits of the theoretical value. However, as Figure 3 suggests these are errors are not noticeable to human perception.

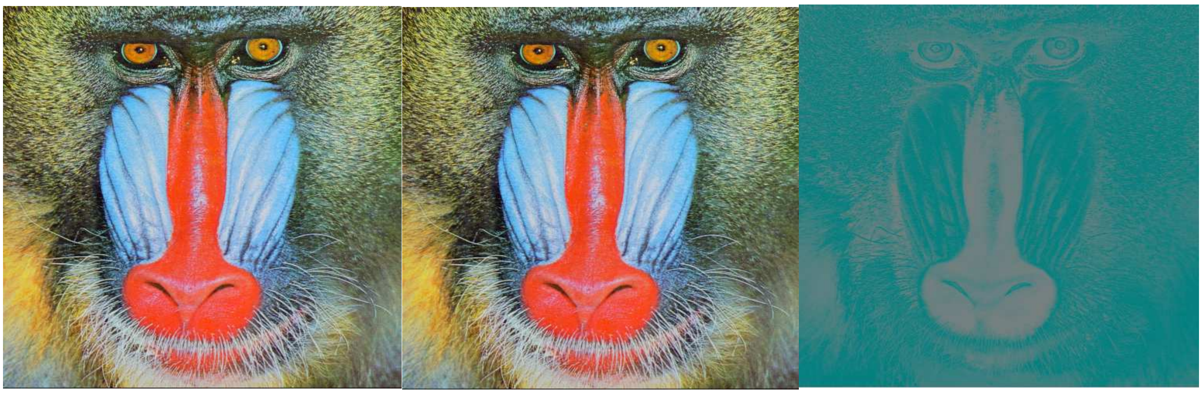

Figure 3: Comparison of an Original Image with a Transformed One

3 Conclusions

A comparison of the approaches presented in this paper and a traditional implementation of the RGB to HSI algorithm showed a significant power saving of 37%. Also the computational throughput of the circuit was improved by a factor of 1.8. This was achieved because in the low-power version of the implementation, a path balancing approach was used, which resulted in a maximum path length of 18ns. The only drawback to the low-power implementation is a higher area requirement of 13%. This was due to the more complex logic required to optimise the switching activity, as well as the additional pipeline stages used to balance the path length.

In conclusion, the task of showing that a thorough investigation at the algorithmic level of the VLSI design cycle can lead to significant power savings was undertaken. As the design was to be mapped to an ASIC, the traditional approach of voltage scaling was not applicable. Therefore, this project has focused on the reduction of the active capacitance in a real-time processing algorithm for the transformation of RGB to HSI. Here various typical design problems such as the division by a constant factor, and the implementation of a pipeline were investigated. Novel approaches have been presented to effectively reduce the active capacitance and therefore the power consumption of the IC. This has resulted in a circuit with a reduced power consumption and increased performance.

Acknowledgement

Andreas Schwarzbacher would like to acknowledge the funding received through the IRCARUS2 (DG-XII) initiative.

References

[1] J. Kender, "Saturation, hue and normalized color," *Carnegie-Mellon University, Computer Science Dept.*, Pittsburgh PA. 1976.

[2] Chandarakasan and S. Sheng, "Low-Power CMOS Digital Design,*" IEEE Journal of Solid State Circuits*, Vol. 27, No. 4, April 1992.

[3] J.E. Volder, “The CORDIC trigonometric computing technique,” *IRE Trans. Electron. Comput.*, vol. EC-8, no. 3, pp. 330-334, Sept. 1959

[4] A.Th. Schwarzbacher, A. Brasching, Th.H. Wahl, and J.B. Foley, "Optimisation and implementation of the arctan function for the power domain," *Electronic Circuits and Systems Conference*, Bratislava, Slovakia, pp. 33-36, September 1999.

[5] A.Th. Schwarzbacher and J.B. Foley, "Optimisation of real-time signal processing algorithms for low-power CMOS implementations," *accepted for Digital Signal Processing 2000*, Bournemouth, Untied Kingdom, July 2000.

[6] A.Th. Schwarzbacher, M. Brutscheck, O. Schwingel and J.B. Foley, "Constant divider structures of the form 2^n±1 for VLSI implementation," *Irish Signals and Systems Conference*, Dublin, Ireland, pp. 368-375, June 2000.