CPRE 488 Homework 2 Spring 2024

Digital Camera Sensor
Jonathan Hess
GitHub Page

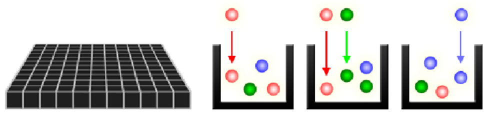
Problem 1

CPU Architecture. Why are ARM processors significantly more prevalent in modern embedded systems than comparable processors from Intel (e.g. Intel Atom-based products)? Provide at least two reasons. What are three meaningful differences between the ARMv8 ISA and x86-64? Be specific.

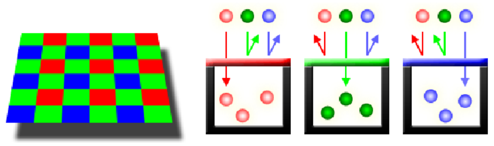
Arm processors do not have to support the entire x86-64 instruction set. The x86-64 is backwards comparable (because porting programs between instruction sets used to be expensive). This means that the instructions set of ARM (RISC: Reduced Instruction Set Computing) is fewer and takes less power. The ARM process is designed for efficiency rather than speed which is optimal for low power control embedded systems.

Problem 2

Digital Camera Sensors. A digital camera uses an array of photo-sensors to capture an image. When the shutter button is pressed, the exposure process begins, and light is directed through a lens to the individual sensors, each of which measures the resulting intensity. The relative quantity of photons at each sensor site are read as electrical signals and collected as digital values, the precision of which is determined by the bit depth.



However, the photosites illustrated above would only be able to capture grayscale images, since the sensors are unable to distinguish between the red, green, and blue intensities of the luminous exposure. To capture color images, a filter has to be placed over each cavity that permits only particular wavelengths of light. Most digital cameras only capture one of the three primary colors at each site, and so they discard (roughly) two-thirds of the incoming light. Consequently, the camera has to approximate the other two primary colors in order to have a full color for each pixel. The most common type of color filter array is called a “Bayer array”, as is shown below.



Problem 2a

The Bayer filter uses a simple strategy: capture alternating red, green, and blue colors at each sensor, with twice as many green filters as red and blue. Briefly explain why more green sensors are allocated than red/blue.

The reasoning behind adding more green than blue or red is to simulate the vision of the eye. "The pattern alternates between red and green light sensitive strips and blue and green strips. The 2:1 ratio of green sensors to red and blue respectively is meant to aid the human eye in correctly seeing colors with the extra data captured by the green sensors." [1]

Problem 2b

Specifically, the Bayer filter pattern is a repeating 2x2 mosaic pattern of light filters, with green filters at opposite corners and red and blue in the other two positions. This pattern means that at any photosite location, one color can be measured directly while the other two will have to be interpolated. This interpolation process (also referred to as "demosaicing") averages the values for the matching neighbor pixels in a 3x3 grid surrounding the center pixel. Provide a C pseudo-code implementation of CFA demosaicing, assuming an input array of 8-bit pixels representing the light intensities of a 1080p image.

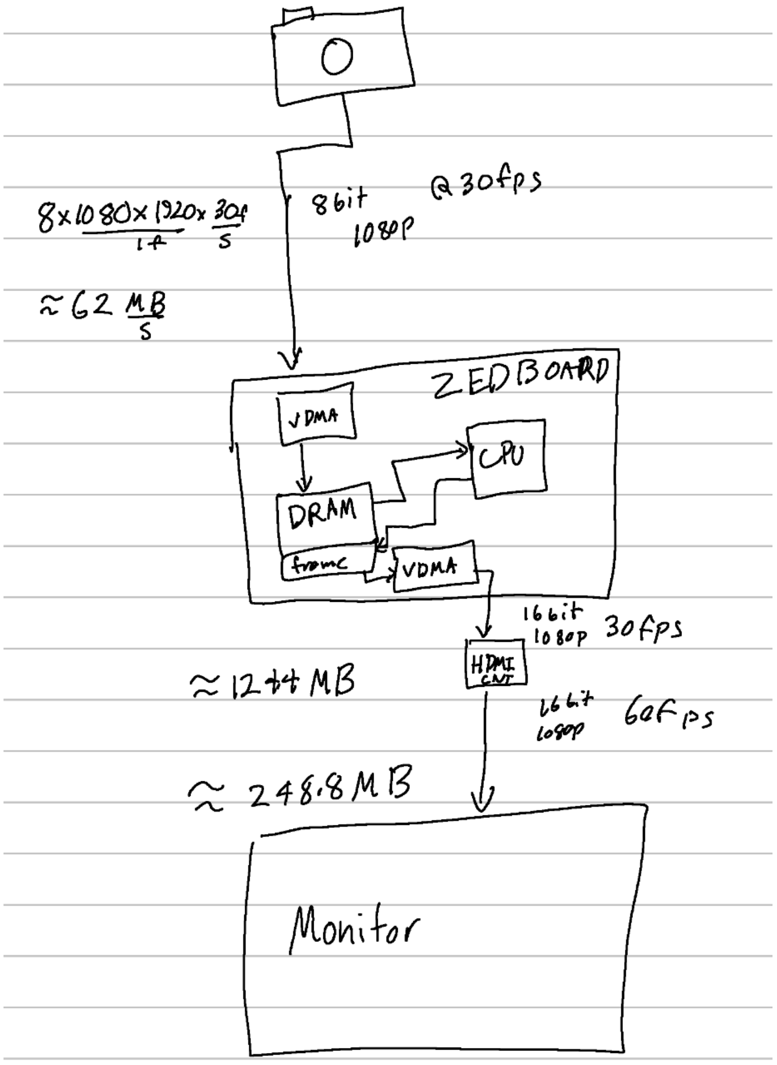
```
//Bayer Filter
Average(image,x,y){
   if((x+y)%2 ==0){ //green
      green = image[x][y];
      red = (image[x][y+1] + image[x][y-1])/2;
      blue = (image[x+1][y] + image[x-1][y])/2;
   }
   else if(x%2==1){ //red
      green = (image[x+1][y] + image[x][y+1] + image[x-1][y] + image[x][y-1])/4;
      red = image[x][y];
      blue = (image[x+1][y+1] + image[x-1][y+1] + image[x-1][y-1] + image[x+1][y-1])/4;
   }
   else{ //blue
      green = (image[x+1][y] + image[x][y+1] + image[x-1][y] + image[x][y-1])/4;
      red = (image[x+1][y+1] + image[x-1][y+1] + image[x-1][y-1] + image[x+1][y-1])/4;
      blue = image[x][y];
   }

   return [reg,green,blue];
}

Average_Image(image){
   output[1920][1080];
   for(y=0; y < 1080; y++){
      for(x=0; x < 1920; x++){
         colors = Average(image,x,y);
         output[x][y] = colors;
      }
   }
   return output;
}
```

Problem 2c

Next, consider the design of an FPGA-based embedded system that implements the main functionality of a digital camera. Specifically, consider the following data processing and memory management steps: • A shutter button triggers light capture through a 1080p sensor array. Assume a high-speed camera that can capture 30 frames per second. • 8-bit grayscale values are streamed sequentially from the sensor array to the FPGA, where the Bayer CFA is first applied, producing 24-bit RGB pixel values. • The resulting 24-bit 1080p image is connected to a Video Direct Memory Access (VDMA) module on the FPGA, which streams the values to a framebuffer DRAM. • A CPU residing on the FPGA reads values from the framebuffer and performs some subsequent processing on the pixels, converting them from a 24-bit RGB format to a 16-bit YCbCr format. These 16-bit pixels are stored in a second framebuffer. • A second VDMA module streams the 16-bit 1080p image to an HDMI controller that is located off- chip. The connected monitor refreshes at a rate of 60 frames per second. Draw a simple diagram illustrating the interconnection between the various components in this camera design. What are the off-chip bandwidth requirements of this system? Be specific, and show your work.



Without compression or other optimization this data throughput would be much harder to achieve in embedded systems. Algorithms like the display stream compression can remove strain from lower level devices.[2]

References

[1] Bayer Color Filter Array

[2] Display Stream Compression