# NanoJPEG CUDA

Parallelization of a JPEG Decompression Library on a CUDA GPGPU

Lorenzo CHIOLA, s287911

# JPEG/JFIF Format

- JPEG: Image Compression
  - choice of **color spaces**
  - choice of **compression quality**
  - choice of **subsampling**
- JFIF: Container Format
  - choice of **sections** (some are mandatory)
  - **progressive JPEG**

# JPEG Compression Overview

- 8x8 px blocks
- Chroma Subsampling
- DCT: Dual Cosine Transform
- Huffman Encoding
- Separate DC component

# JFIF (JPEG File Interchange Format)

- **SOF** (Start Of Frame)
  Width, height, color space...
- **DHT** (Define Huffman Table)
  Huffman code tree. Y DC, Y AC, Cb/Cr DC, Cb/Cr AC
- **DQT** (Define Quantization Table)
  Separate for Y and Cb, Cr
- **DRI** (Define Restart Interval)
  DC DPCM restart component
- (Start of) **Scan**
  Actual data: 8x8 blocks

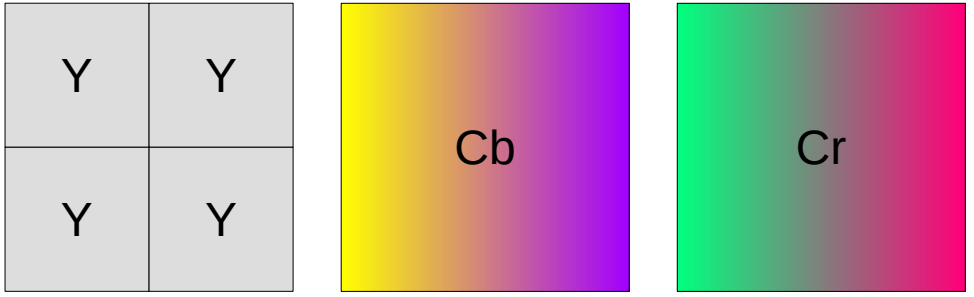
# JFIF Scan

- **8x8 Blocks** grouped into MCUs (Minimum Coded Units)

```
RGB:           {R, G, B},   {R, G, B},   …
YcbCr 4:4:4:   {Y, Cb, Cr}, {Y, Cb, Cr}, …
```

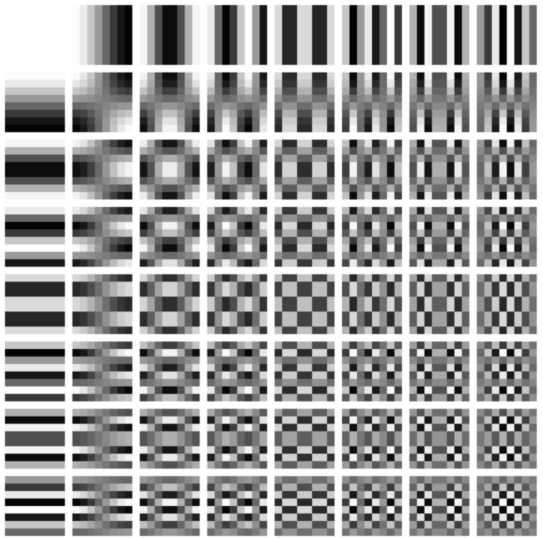
- **Chroma Subsampling** (less chrominance information)

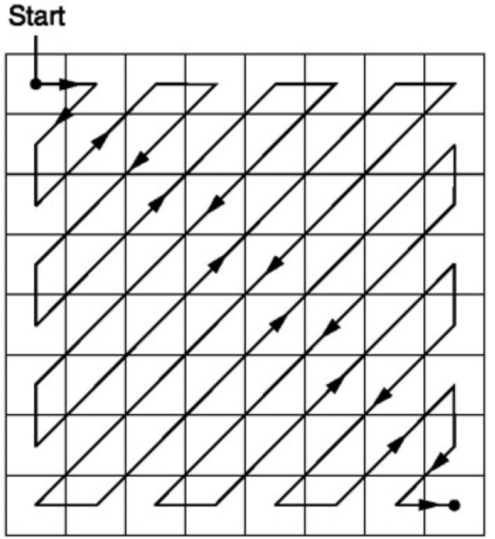
```
YcbCr 4:2:2:   {Y, Y, Cb, Cr},       {Y, Y, Cb, Cr},       …
YcbCr 4:2:0:   {Y, Y, Y, Y, Cb, Cr}, {Y, Y, Y, Y, Cb, Cr}, …
```



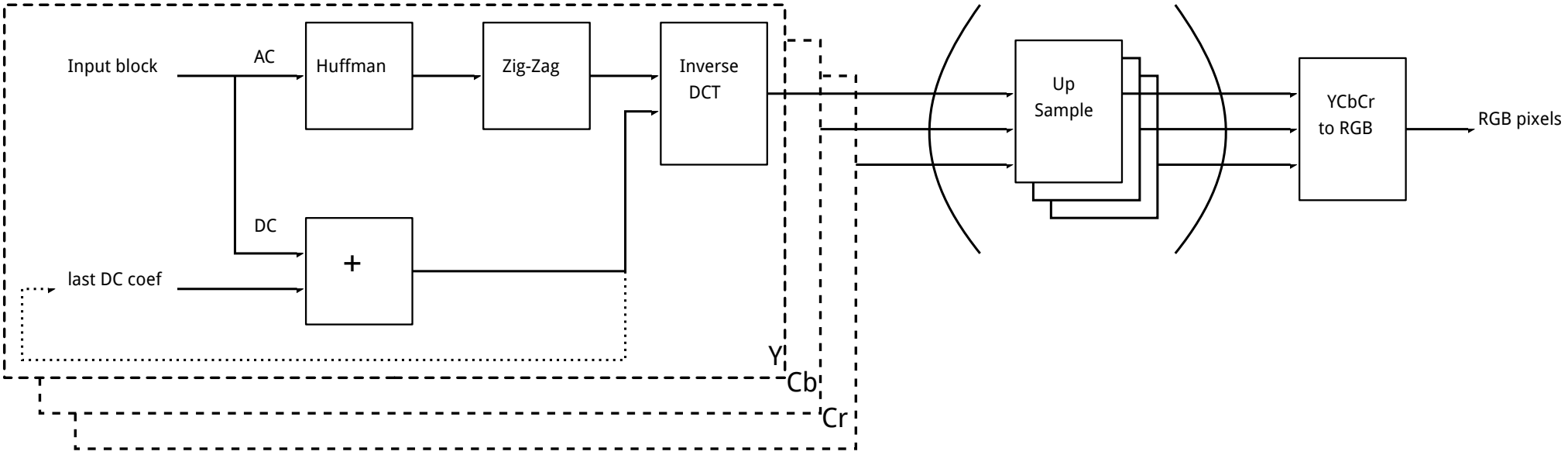
# JFIF Scan: Blocks

- **DC coefficient** (zero frequency: cell [0, 0])
  Differential Pulse Code Modulation (DPCM)
- **AC coefficient** (the other 63 cells)
  Zig-Zag order: from low to high frequency
  Huffman encoded

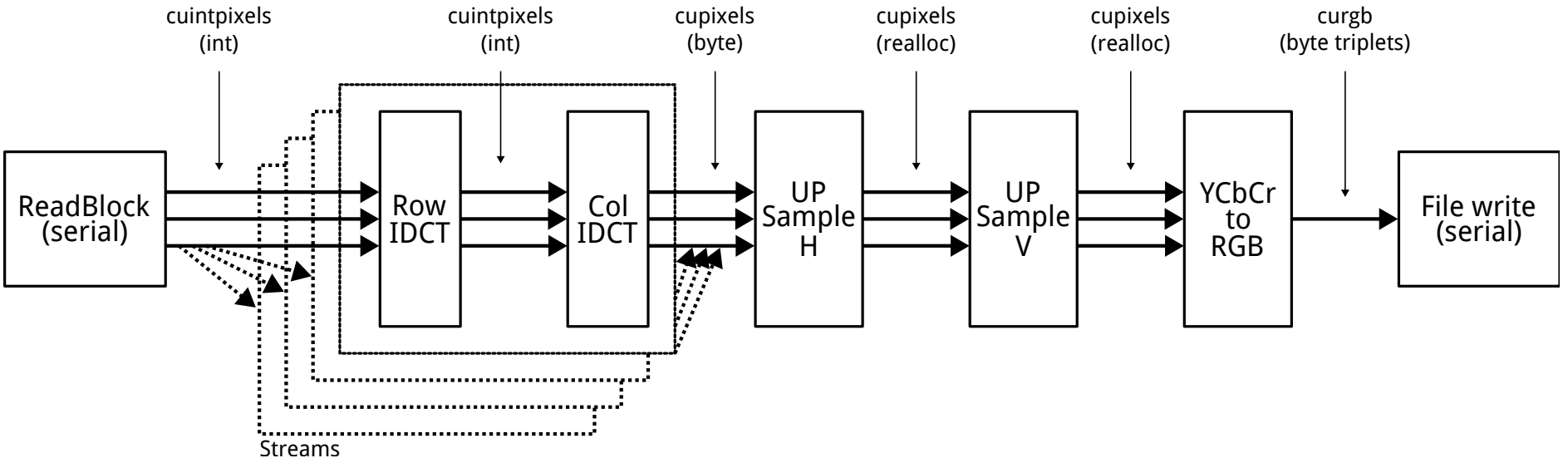




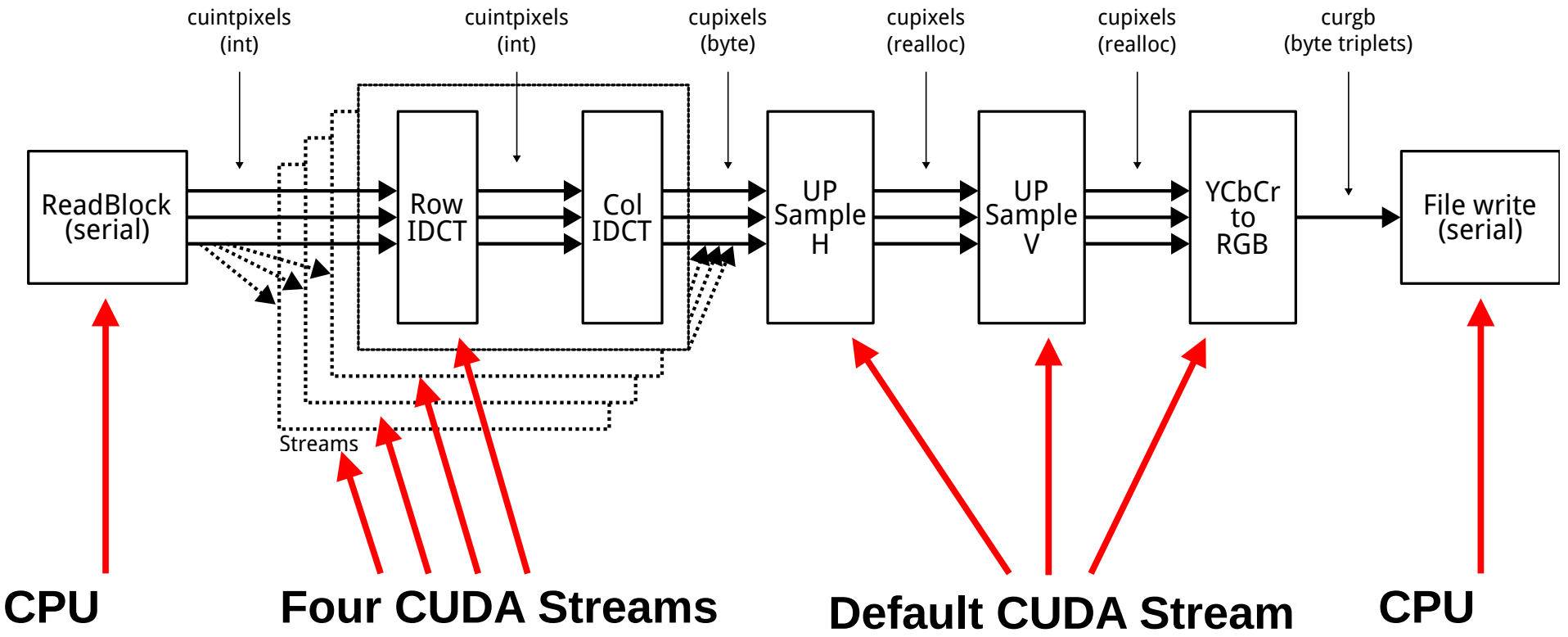
# JPEG compression flow



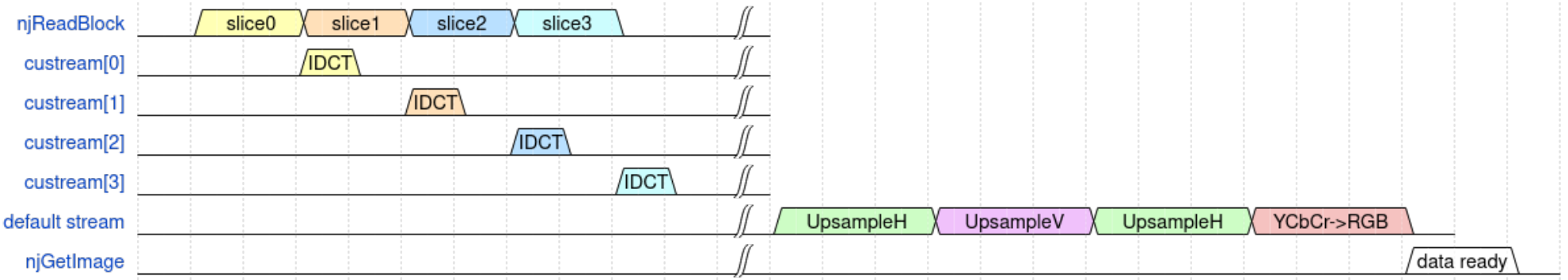
# NanoJPEG CUDA Data Flow



# Streamed execution

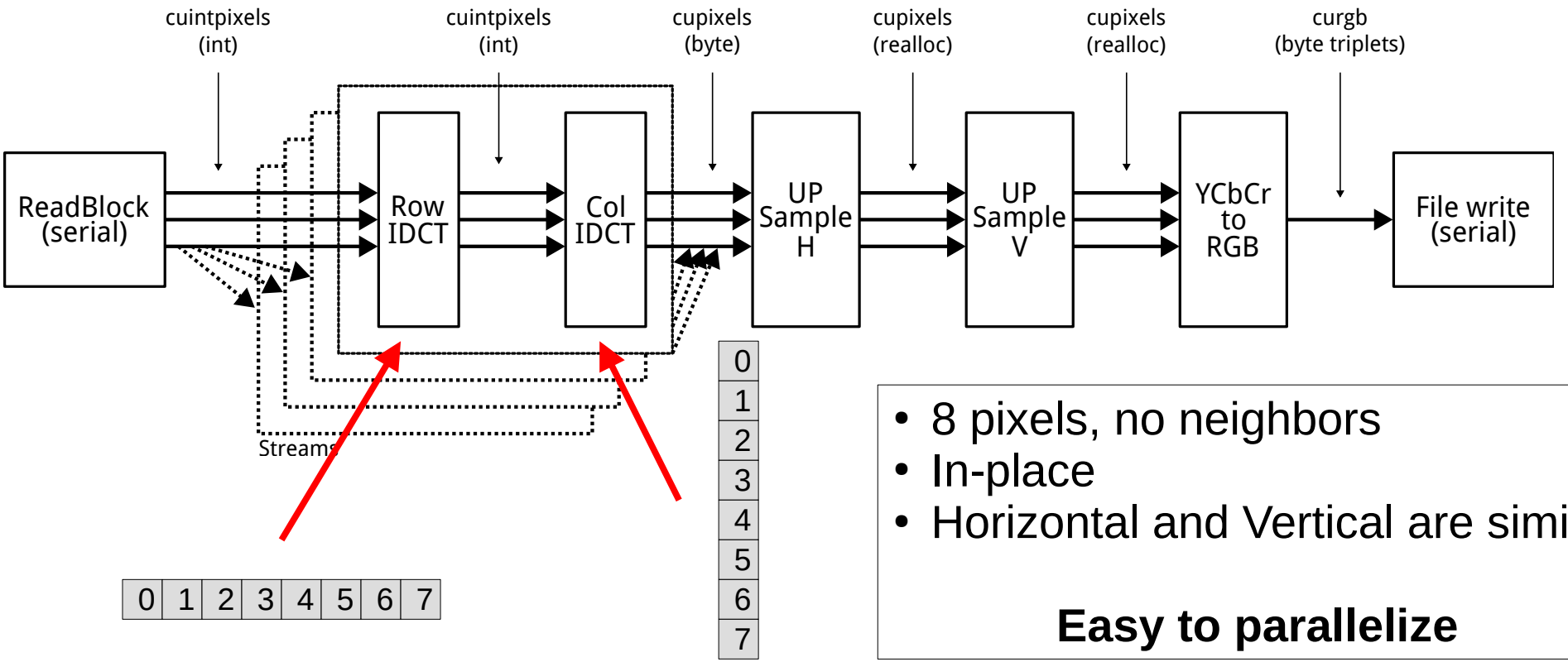


# Streamed execution




- Few CUDA cores: 1 SM / 128 Threads
- Disk I/O bound workload

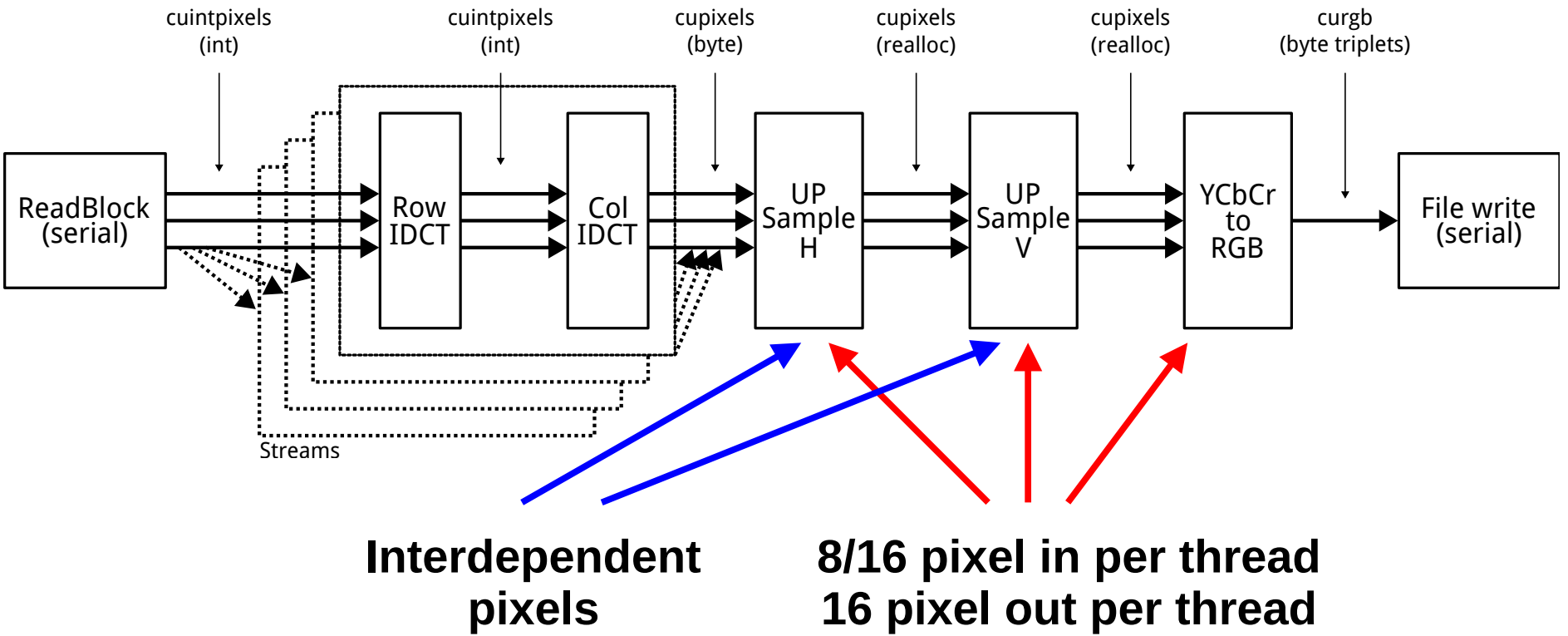
# Inverse Discrete Cosine Transform (IDCT)




- 8 pixels, no neighbors
- In-place
- Horizontal and Vertical are similar

**Easy to parallelize**

# Upsampling, RGB conversion



# NanoJPEG CUDA Performance

- Advantageous for big images (on Jetson Nano)
  1 Megapixel: GPU **on par** with single-thread CPU
  8 Megapixel: GPU **40% faster** than single-thread CPU
- Pro: elaboration in parallel with disk read
- Problems: elaboration is faster than disk I/O (bottleneck)

Performance must be evaluated
for any particular CPU + GPU + DISK combination