OpenCL Tutorial

David Castells-Rufas
Microelectronics & Electronics Systems Department
Universitat Autònoma de Barcelona
david.castells@uab.cat

Centre de Prototips i Solucions Hardware - Software
Center for Hardware – Software Prototypes & Solutions

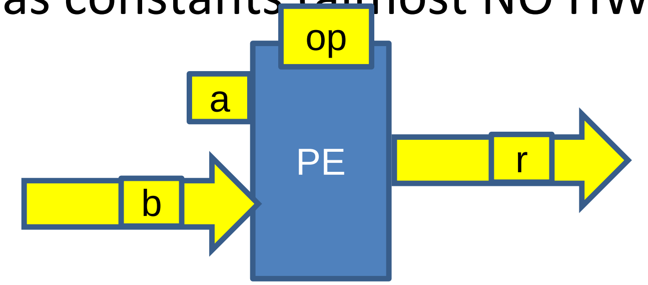
Optimizing

Goals

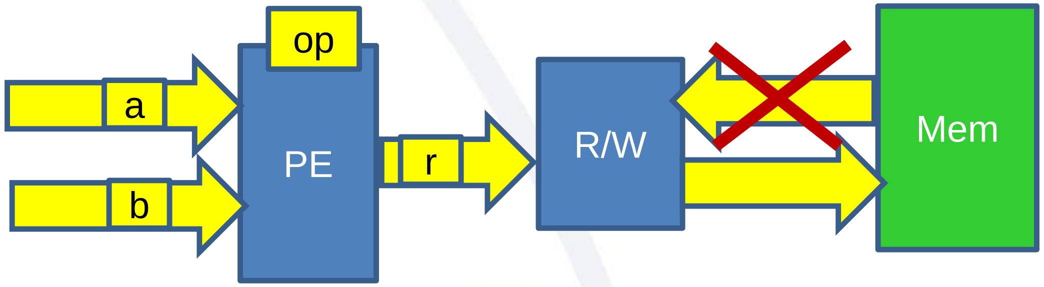
- Work on kernels to detect bottlenecks and try to optimize them
- LAB3.1
- LAB3.2
- LAB3.3

Improvement: Constant variables

- __constant
  - –Ensures that variables are used as constants (almost NO HW COST)

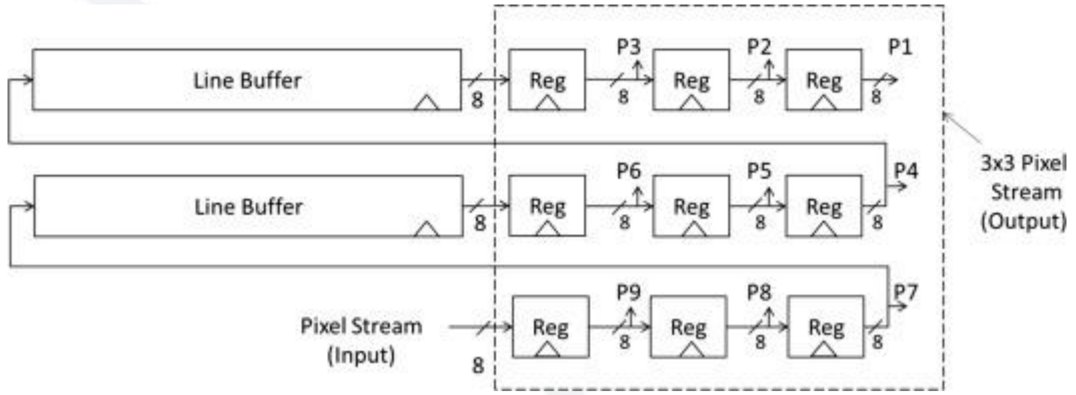


- __restrict
  - –Ensures that only a Load or Store unit is implemented for a memory region



Line Buffers

- Implement a line buffer to reduce memory access / 9
- Common hardware design strategy for image processin



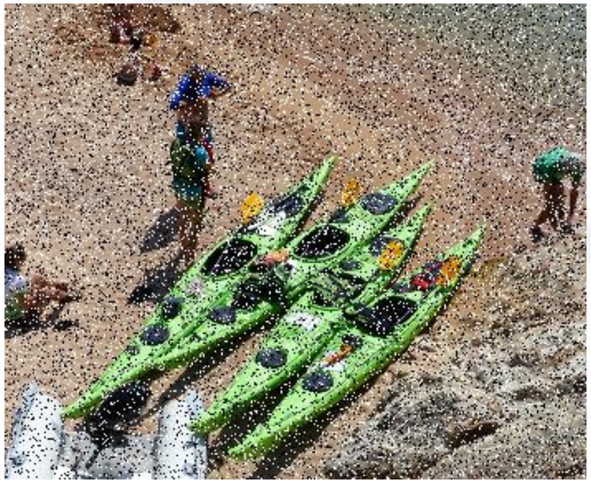
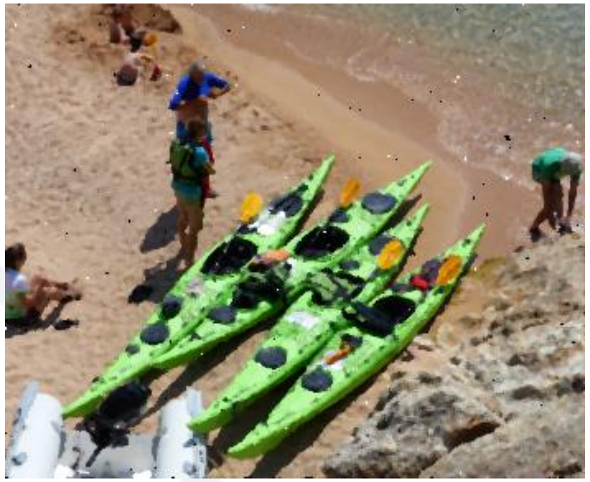
- A line buffer is a shift register! **We can express it using OpenCL!**
- **We must use a Pipelined Kernel Design (Single WorkItem)**

Inferring Line Buffer

```
for (int i=-FIFO_SIZE; i < w*h; i++)
  {
      #pragma unroll
      for (int k=0; k < FIFO_SIZE-1; k++)
      {
          fifo_r[k] = fifo_r[k+1];
          fifo_g[k] = fifo_g[k+1];
          fifo_b[k] = fifo_b[k+1];
      }

      int xr = (i+FIFO_SIZE) % IMAGE_WIDTH;
      int yr = (i+FIFO_SIZE) / IMAGE_WIDTH;

      unsigned char r, g, b;
      image_getRGB(inputImage, w, h, xr, yr, &r, &g, &b);

      fifo_r[FIFO_SIZE-1]= r;
      fifo_g[FIFO_SIZE-1]= g;
      fifo_b[FIFO_SIZE-1]= b;
```

LAB3.1

- Median Filter

- What we want:

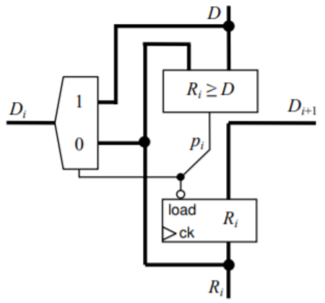


Fig. 4. Data-slice module of the insert sort register file.

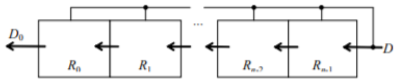


Fig. 5. Linear composition of n data-slice modules.

LAB 3.2

- What we want

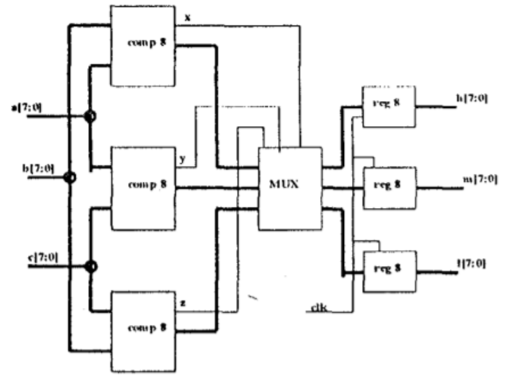


Figure 2. Block schematic of sorter

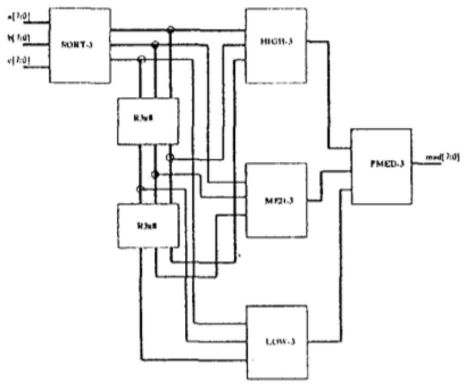
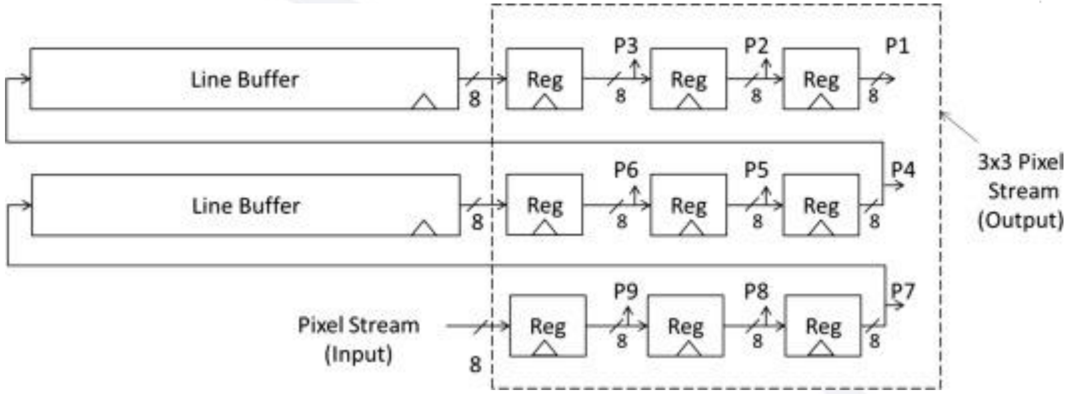


Figure 3. Top level schematic of median filter

LAB 3.3

- What we want



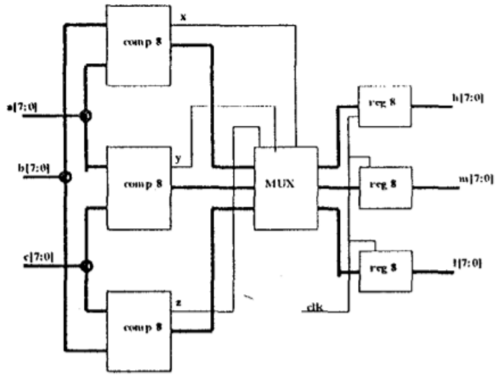


Figure 2. Block schematic of sorter

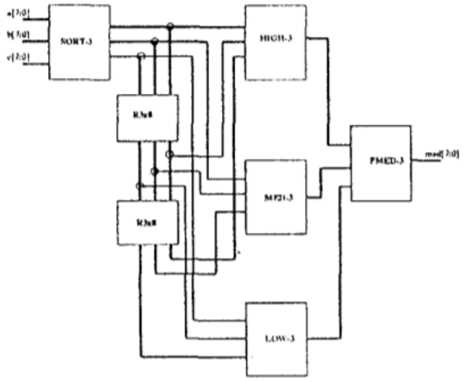


Figure 3. Top level schematic of median filter