

# Organización del Computador II

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

## Trabajo Práctico Número 2

**Grupo: Tu me pixeleas**

Integrante	LU	Correo electrónico
Costa, Manuel José Joaquín	035/14	manuc94@hotmail.com
Gatti, Mathias Nicolás	477/14	mathigatti@gmail.com
Pondal, Iván	078/14	ivan.pondal@gmail.com

## 1. Introducción

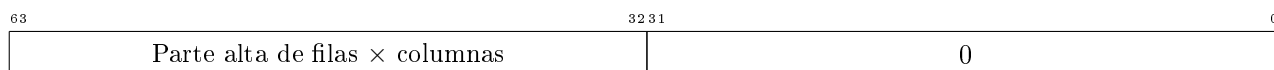
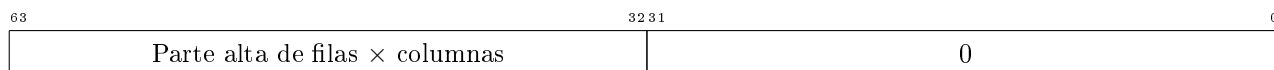
## 2. Desarrollo

### 2.1. Explicación del código

#### diff\_asm

Como primer paso computamos el total de píxeles que tiene la imagen. Para esto simplemente realizamos el producto entre la cantidad de filas (en `eax`) y de columnas (en `r8d`). Como es un producto de enteros de 32 bits, según nos dice el manual, la parte baja del resultado se guarda en `eax`, mientras que la alta lo hace en `edx`. Ya que lo que queremos es tener el resultado entero en `rcx`, realizamos un shifteo lógico de `rdx` que deja la parte baja del mismo con 0's y la parte alta con lo que solía tener la parte baja.

Figura 1: Suma de registros



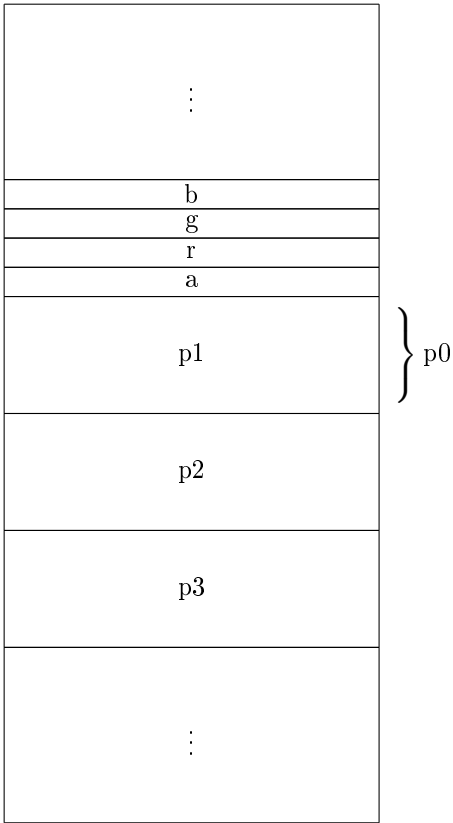
Dado que cada pixel son 4 bytes, y las instrucciones SIMD me permiten operar con 16, vamos a procesar de a 4 píxeles por iteración. Por lo tanto dividimos por 4 `rcx`.

A continuación analizamos una iteración del ciclo:

Primero, cargamos 4 píxeles de `src` y `src2` en `XMM1` y `XMM2` respectivamente. Es importante recordar que al cargar los píxeles en cada registro, el orden relativo de los mismos es el inverso al que tenían en memoria, así como el de sus componentes. Para ilustrar mejor esta idea puede verse la figura (????).

Luego, queremos realizar restas componente a componente (byte a byte), pero como existe la posibilidad de que den resultados negativos tenemos que extender los operandos de bytes a words para poder representar los números entre -255 y 255. A tal fin usamos las operaciones de desempaqueado de byte a word, y extendiendo las componentes de los píxeles con un cero (pues todas son positivas).

Figura 2: Copiado de cuatro píxeles en un registro XMM



63	3231	0
Parte alta de filas × columnas		0

3. Resultados

## 4. Conclusiones