



## **Sistemas Digitales**

(66.17/86.41)

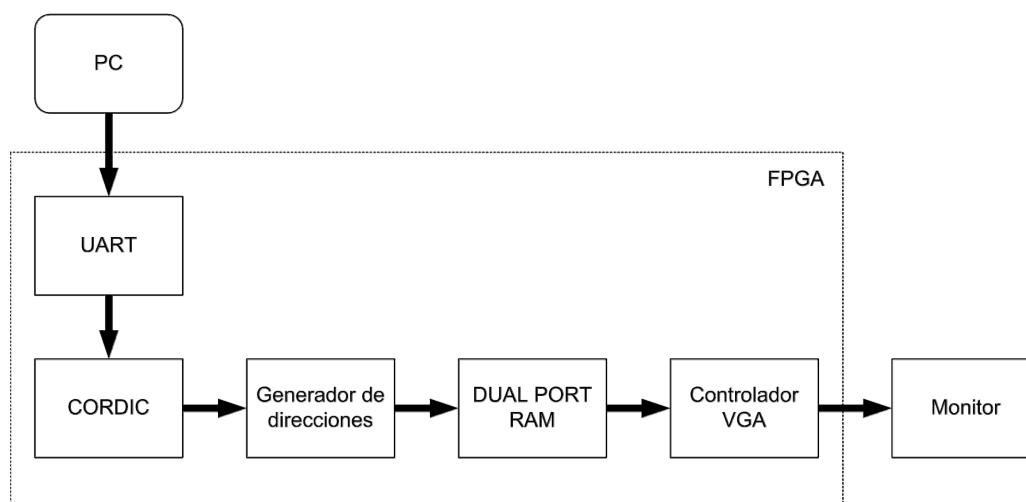
Trabajo práctico final  
(Versión C)

Diseño de un motor de rotación gráfico  
2D basado en el algoritmo CORDIC

## 1. Objetivo

En el presente Trabajo Práctico el alumno desarrollará una arquitectura de rotación de un vector en 2D, basada en el algoritmo CORDIC. El objetivo principal es desarrollar tanto la unidad aritmética de cálculo como así también el controlador de video asociado a la visualización del movimiento.

Para la realización completa del trabajo práctico se utilizará una interfaz serie UART, por medio de la cual se comandará el giro del vector. A partir de los valores de las componentes, se rotará el vector en el plano XY según el valor que adquieran las entradas del sistema, y por último las componentes rotadas serán presentadas en un monitor VGA. En la *Figura 1* puede observarse un diagrama en bloques del sistema completo. Deberá determinarse la cantidad mínima de bits de ancho de palabra (bits de precisión) para alcanzar las especificaciones requeridas.



*Figura 1. Esquema completo del sistema*

## 2. Desarrollo

La representación de las coordenadas 2D de los puntos que forman el vector se deja a criterio del alumno.

Para lograr la rotación del vector se deberán enviar comandos a través de la UART, respetando el siguiente formato:

*ROT C D* (indica rotación **C**ontinua en sentido **H**orario o **A**nti-horario, que se establece por medio del parámetro **D**)  
 Ejemplo: *ROT C H* / *ROT C A*

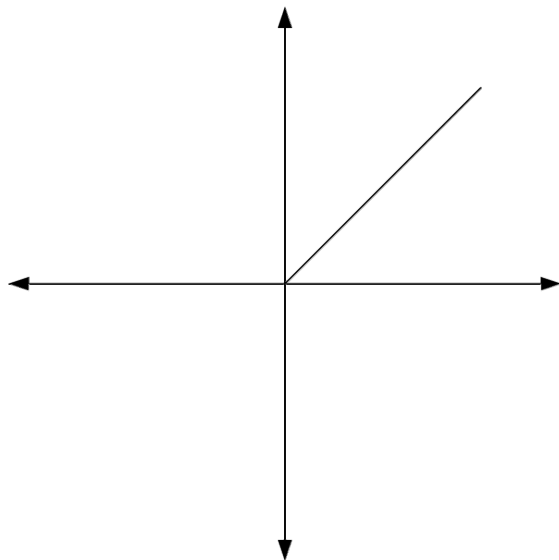
*ROT A ang* (indica la rotación del vector un ángulo *ang*)  
 Ejemplo: *ROT A 45* (se indica rotar el vector 45°)

Los datos de salida del rotador de CORDIC deberán ser almacenados en una memoria (memoria de video) que deberá ser leída por un controlador (controlador de video) para mostrar la imagen en pantalla. La calidad de video será VGA monocromático de 1 bit de 640x480 pixeles. De esta forma, las coordenadas (x,y) de cada punto de la imagen corresponderán a una dirección de la memoria de video, donde se escribirá un 1 lógico en el caso en que en dicha posición de memoria se encuentre un punto de la misma. Por lo tanto, cada coordenada deberá ser mapeada a una dirección de memoria. Si se utilizara toda la pantalla la memoria debería constar de 307200 (640x480) posiciones de 1 bit cada una. Como

los ejes de coordenadas y el vector pueden hacerse de un tamaño menor al tamaño total de la pantalla no resulta necesario almacenarla completamente. Queda a criterio del alumno establecer el tamaño en el que se verá la rotación del vector.

Debido a que el controlador VGA y el rotador escriben y leen a la vez direcciones de memoria de video que son independientes, la memoria de video deberá ser implementada por medio de una DUAL PORT RAM, utilizando para ello las Block RAMs disponibles en la FPGA.

Notar que luego de producirse una rotación del vector se deberá limpiar la memoria de video. Caso contrario, aparecerán en pantalla la imagen actual y la precedente a la rotación. Por lo tanto, entre dos rotaciones sucesivas debe realizarse un ciclo de limpieza de la memoria de video.



*Figura 2. Visualización en pantalla del vector y los ejes de coordenadas*

### 3. Especificaciones

- Resolución de video: 640 x 480 1 bit monocromo 50 Hz.
- Velocidad angular de rotación mínima: 35.15625 grados/seg.
- Paso angular  $\Delta\phi$ : 0.703125 grados.
- Dispositivo: Spartan 3E-500 (Kit Nexys 2 Board o Starter Kit Board).

### 4. Entregables

Para la aprobación del trabajo práctico se debe entregar:

1. Informe en el que se detalle cada una de las etapas de diseño de la arquitectura implementada (diagramas en bloque, ecuaciones lógicas, criterios adoptados en la toma de decisiones, capturas de pantalla de simulaciones relevantes, conclusiones finales)
2. Código VHDL de la arquitectura implementada.
3. Códigos auxiliares (C, Matlab, etc).