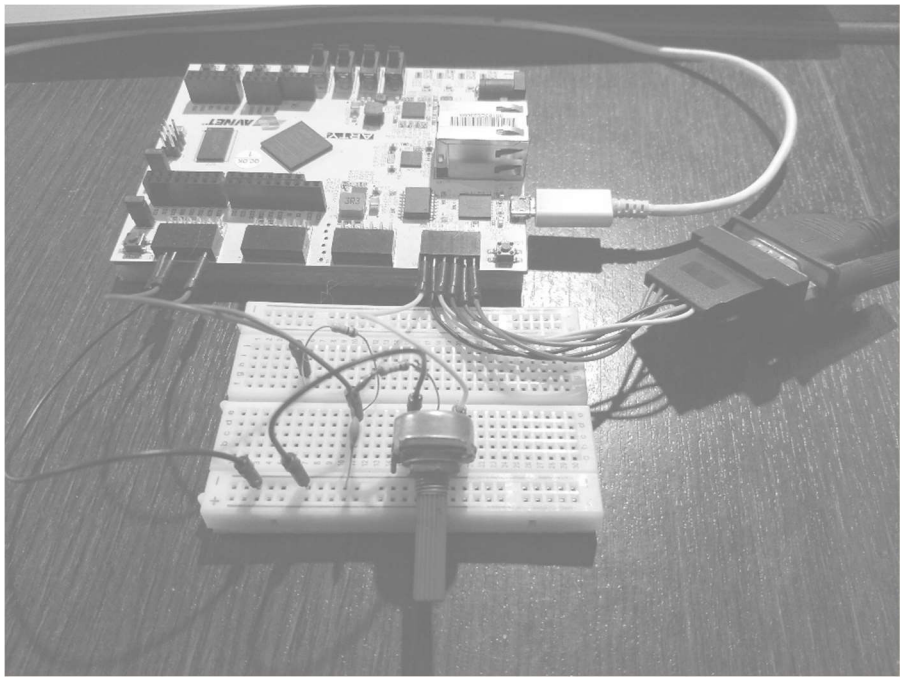
Electrónica Digital 1

Trabajo Práctico Final
“Voltímetro digital con salida VGA”



Estudiante: Rodriguez, Javier Ceferino

Profesores: Álvarez, Nicolás

Sagreras, Miguel Ángel

Periodo de cursada: 1° Cuatrimestre 2020

1 - Objetivo

Para el trabajo final, se deberá diseñar un Voltímetro Digital con salida VGA, con el fin de poder observar las mediciones hechas a través de un monitor. Para llevar a cabo esto, se debe describir la arquitectura, realizar simulaciones, sintetizar e implementar en una placa FPGA (Field Programmable Gate Array) Arty A7-35 de Xilinx Digilent.

2 - Resumen

En este informe, se describirán las funciones de cada bloque y sus jerarquías, se muestran mediciones y simulaciones realizadas (tanto en Vivado como en la práctica real). También se mostrarán los criterios de diseño implementados, antes y después de las pruebas, y algunas tablas elaboradas, y adquiridas al realizar la síntesis e implementación de la programación.

3 - Desarrollo

3.1 - Consideraciones

Para poder implementar un dispositivo capaz de medir un rango de tensión de 0.00V a 3.30V, se tuvo en cuenta las siguientes restricciones de diseño:

Las herramientas clasificadas como **comportamiento** no se pueden usar. Estas son:

- if-then-else
- case-when
- for-while-loop-exit-next

Exceptuando el **process**, se pueden usar todas las instrucciones llamadas **concurrentes**.

En cuanto a visualización del voltaje medido, se deben mostrar 5 dígitos BCD: Uno para la parte entera, uno para el punto, dos para decimales y el último para la unidad del volt.

3.2 - Diseño

Partiendo del diseño base propuesto (Figura 1 y Figura 2), se implementan los códigos en idioma VHDL, teniendo en cuenta las restricciones anteriormente mencionadas.

Para verificar que el voltímetro funcione correctamente, se implementa un potenciómetro a la entrada para generar un barrido de tensiones. Debido a no se ha considerado la impedancia de entrada de los PMOD, existen una tensión de offset de 0.60V aproximadamente.

Como se mencionará en *3.5 - Primeras pruebas con FPGA*, se tuvo que incrementar el margen de conteo de bits debido a la inestabilidad de la medición.

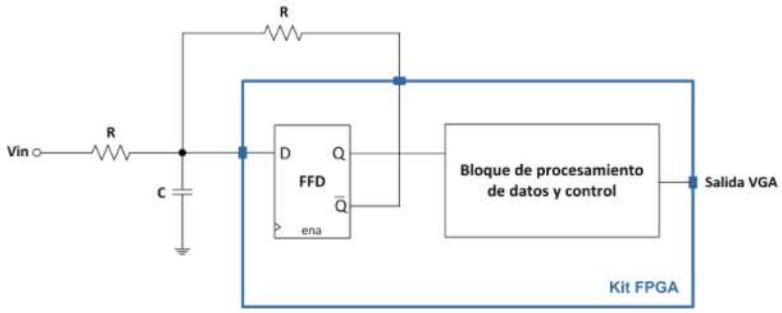


Figura 1: Diagrama en bloques del dispositivo a implementar

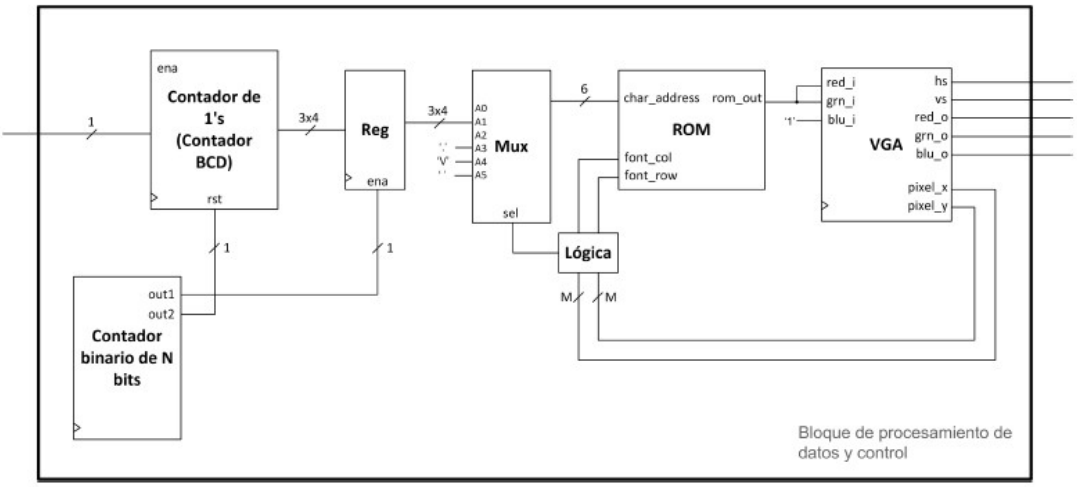


Figura 2: Diagrama en bloques del Bloque de procesamiento de datos y control a implementar

A continuación, se describirán los componentes desarrollados a nivel software.

Flip Flop D (ffd.vhd): Biestable utilizado en la mayoría de los componentes del voltímetro para recrear registros, contadores, sincronismos y demás.

Conversor Analógico Digital Sigma-Delta (ADC_sd.vhd): Es uno de los componentes más importantes del voltímetro. Con ayuda de la parte analógica (resistores, capacitor), toma la señal de entrada a modo de poder comparar y obtener en la salida una serie de unos y ceros, dándonos una estimación de que tensión existe en su entrada. La salida afirmada (Q), envía estos datos binarios al Bloque de procesamiento de datos y control. La salida negada (/Q), se realimenta para realizar la comparación en la entrada.

Registro de N bits (reg_Nb.vhd): Este registro es básicamente la conexión en paralelo de N Flip Flops D, y con esto es posible guardar N datos binarios. Se utiliza principalmente en todos los contadores.

Contador BCD (BCD_counter.vhd): Como lo dice su nombre, es un contador encargado de generar la **secuencia BCD** (del 0 al 9, en binario). Está pensado para dar la información de la cuenta que lleva (sincronizada por reloj) a través de **count**, y de avisar en **max** si se llegó a su cuenta máxima. Utiliza un registro de 4 bits, un sumador y un comparador.

Contador de unos (cOnes.vhd): Utilizando 7 contadores BCD en cascada (originalmente eran 5), en una ventana de tiempo establecida, se cuenta hasta cierto valor. Se toman los 3 dígitos más significativos que corresponden al voltaje medido en la entrada. Cada salida es de 4 bits.

Contador binario (c33k.vhd): Este contador realiza un incremento desde **0** hasta **3300000**. Es el encargado de indicar cuando se debe habilitar el registro que guarda el dato del voltaje medido y cuando debe resetear la cuenta del contador de unos. Utiliza un registro de 22 bits, un sumador y dos comparadores.

Sumador 1 bit (sum_1b.vhd): Este realiza la suma de dos datos binarios, teniendo en cuenta el carry de entrada. Su salida es la suma con carry.

Sumador de N bits (sum_Nb.vhd): Sumadores de 1 bit interconectados de tal forma que dé la suma de valores de N bits.

Comparador de N bits (comp_Nb.vhd): Realiza la comparación entre dos valores de N bits. En su salida hay un 1 si la comparación es verdadera.

Multiplexor de 4 bits (mux.vhd): Es un multiplexor pensado para recibir los datos de 4 bits guardados en el registro de voltaje. En 2 de sus entradas, están harcodeados los valores correspondientes para el punto y la unidad. La selección de estas entradas está controlada por los 3 bits más significativos del **pixel X** de la VGA.

ROM de caracteres (ROM.vhd): Esta ROM tiene almacenado mapeos de binarios para dibujar los caracteres en el monitor (0 al 9, punto y unidad). Para el mapeo de cada pixel está comandado por la **dirección** que recibe desde el multiplexor y el **pixel X** (bits del 4 al 6) y **pixel Y** (bits del 4 al 6) de la VGA. Se usan como coordenadas pixel X y la unión entre la dirección y pixel Y. La salida de la ROM indica cuando debe prender o apagar cada píxel.

Contador horizontal (cHor.vhd) y contador vertical (cVer.vhd): Son contador que se utilizan para poder tener en cuenta el back porch, front porch, parte visible y un pulso. En cada contador, se realiza la cuenta correspondiente al valor total de la señal de sincronismo (800 píxeles para el horizontal y 525 líneas para el vertical).

VGA (VGA.vhd): En este bloque, se realiza el control de colores de cada pixel a través de sincronismos e indica al resto de los bloques que dígito se debe graficar en el monitor (a través de sus salidas pixel X y pixel Y). En este caso, se desarrolla para una resolución de 640x480. Las otras salidas, rojo, verde, azul, sincronismo horizontal y sincronismo vertical, se conectan a un cable para VGA.

Voltímetro (Voltimetro.vhd): Acá se realiza la conexión de la mayoría de los bloques, mínimo necesario para poder hacer las primeras simulaciones con testbenchs en Vivado y/o ModelSim.

Voltímetro Top Level (Voltimetro_Esquema_Arty_A7-35_top_level.vhd): Plantilla que contiene el voltímetro y un MMCM generado en Vivado (clocking wizard). Contiene los datos necesarios para declarar que pines de los PMOD del Arty A7-35 se utiliza.

Los códigos desarrollados están en la carpeta “Codigos”.

3.3 - Jerarquía

En la Figura 3, se ve como está conformado el voltímetro a nivel jerárquico. También se indican cuantos componentes se usan dentro de otro

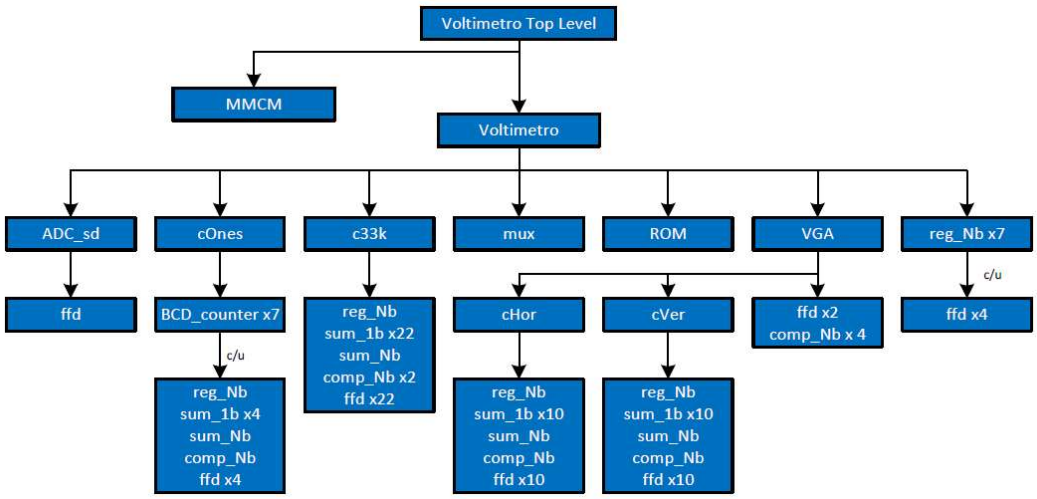


Figura 3: Jerarquía de componentes.

3.4 - Simulación

ModelSim: En este programa, se realizaron las simulaciones principales. Una de ellas, y la más importante, es la **prueba 0**.

En la prueba 0, se puede apreciar un primer vistazo la habilitación que genera la VGA para que se puedan ver los pixeles en la parte visible de cada señal de sincronismo.

En la Figura 4a y 4b, se ven las señales de sincronismo horizontal, vertical, los colores y el clock.

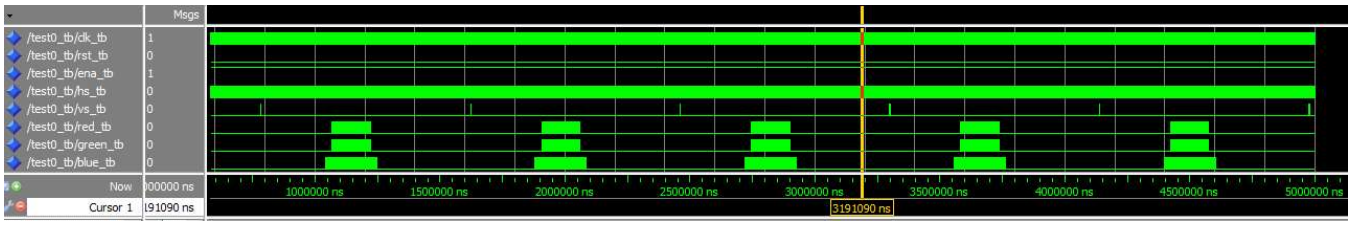


Figura 4a: Simulación de prueba 0 en 5 ms.

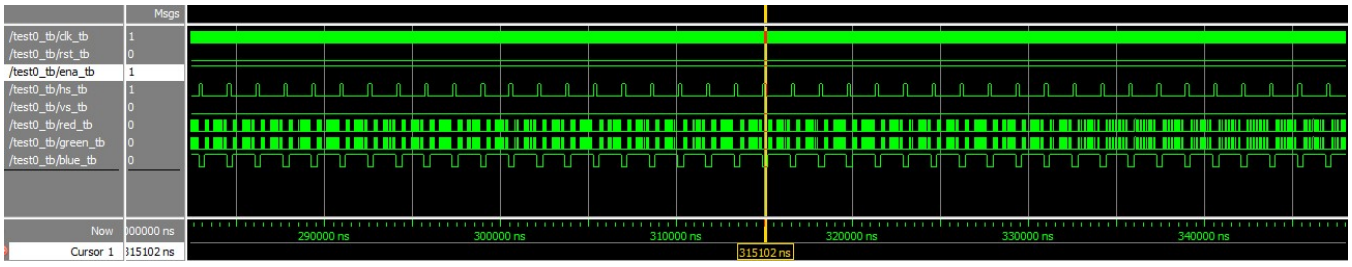


Figura 4b: Simulación más detallada en partes visibles de sincronismos.

Vivado: En este otro programa, que se usa para programar la FPGA, se generan tablas de utilización y consumo (Figura 5 y 6). Acá se ve claramente que Vivado optimiza el proyecto creado al eliminar Flip Flops que no se utilizan (registro de voltaje, se usan solamente 12).

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (210)	BUFGCTRL (32)	MMCME2_ADV (5)
N Voltimetro_toplevel	116	85	2	39	116	57	9	2	1
clk25MHz_gen (clk_ge...	0	0	0	0	0	0	0	2	1
inst_voltimetro (Voltim...	116	85	2	39	116	57	0	0	0
ADC (ADC_sd)	1	1	0	1	1	0	0	0	0
contBin (c33k)	34	22	0	11	34	19	0	0	0
contUnos (cOnes)	32	28	0	9	32	20	0	0	0
Monitor4K (VGA)	49	22	0	18	49	18	0	0	0
contH (cHor)	22	10	0	8	22	9	0	0	0
contV (cVer)	27	10	0	10	27	9	0	0	0
ffdH (ffd_76)	0	1	0	1	0	0	0	0	0
ffdV (ffd_77)	0	1	0	1	0	0	0	0	0
reg_gen[3].regNx (r...	0	4	0	2	0	0	0	0	0
reg_gen[5].regNx (r...	0	4	0	2	0	0	0	0	0
reg_gen[6].regNx (r...	0	4	0	2	0	0	0	0	0
ROMcito (ROM)	0	0	2	2	0	0	0	0	0

Figura 5: Tabla de utilización (desde Implementación). Se ven los LUTs, Slices y Flip Flops utilizados.

Resource	Utilization	Available	Utilization %
LUT	116	20800	0.56
FF	85	41600	0.20
IO	9	210	4.29
MMCM	1	5	20.00

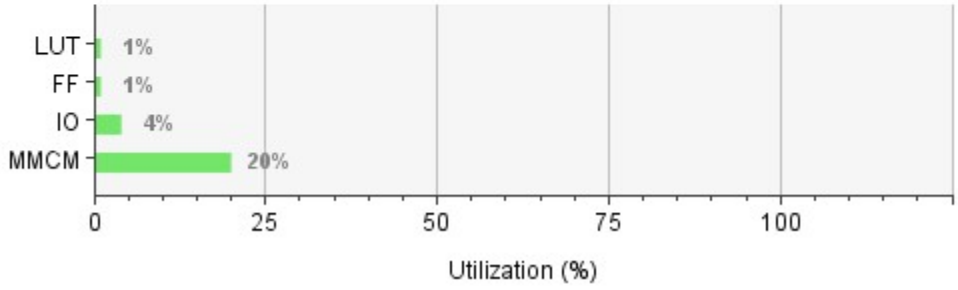


Figura 6: Tabla y gráfico de utilización de LUTs, Flip Flops, Entradas/Salidas y MMCM.

3.5 - Primeras pruebas con FPGA

Luego de realizar la prueba 0, inmediatamente se pasó a la síntesis, implementación y generación de archivos para poder llevar a cabo la prueba práctica.

Lo que se ha visto primero fue que existe un offset de 0.60V aproximadamente y se concluyó que es debido a que el circuito analógico no se adaptaba a la impedancia que existe en los PMOD.

Lo segundo que se observó fue que la medición no era estable (había variación de ± 0.15V). Esto no permitía ver claramente que tensión se estaba midiendo en la entrada. Como solución, se propuso incrementar la ventana de tiempo, para que los contadores puedan permitir adquirir la medición más exacta posible (teniendo en cuenta la tensión de offset). Este incremento se lograba agregando dos contadores BCD, siendo ahora en total 7, y que el contador binario cuente hasta 3300000.

Luego, se adaptó un potenciómetro (100KΩ), para lograr tener un barrido de tensiones de 0.60V hasta 3.29V.

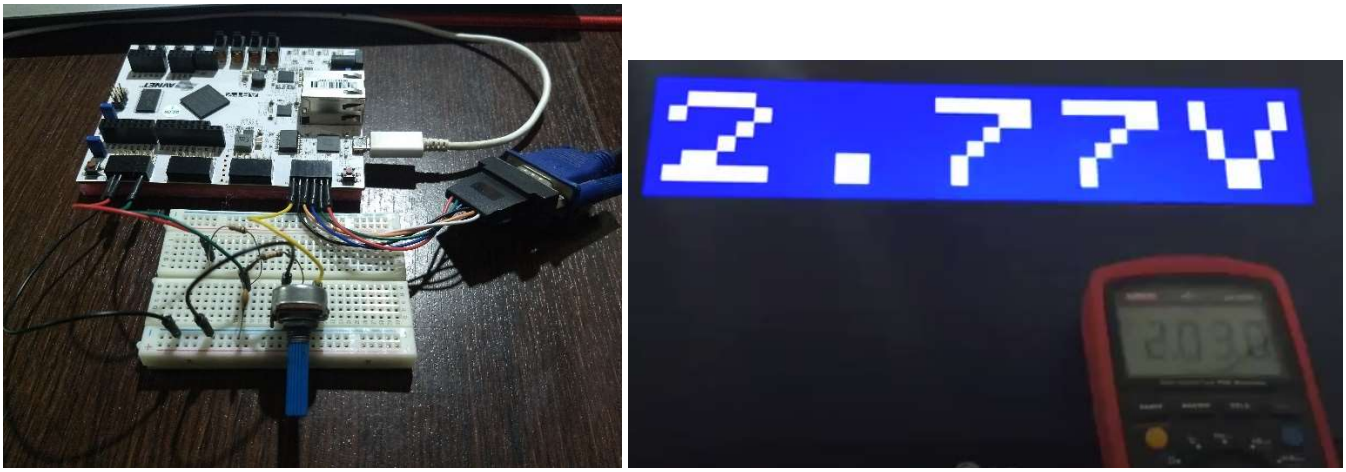
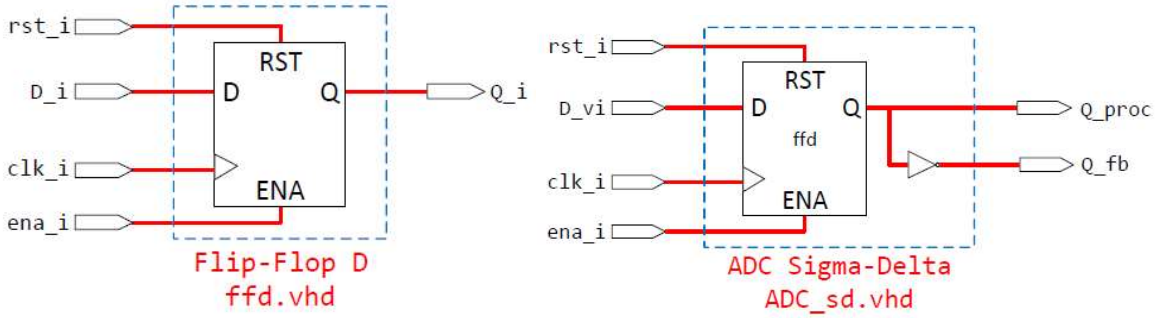


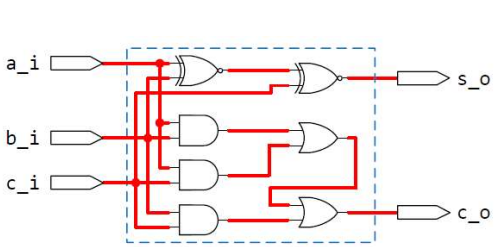
Figura 7: FPGA con circuito propuesto, con implementación de potenciómetro y adaptador VGA casero. Prueba en monitor y tester.

En la carpeta “Archivos”, se encontrará un video de la simulación.

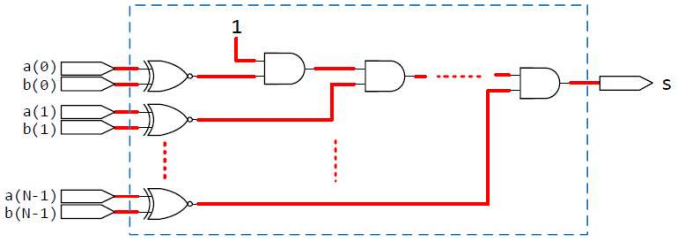
4 - Circuitos

Para terminar, se muestran todos los componentes desarrollados, con sus respectivas entradas y salidas.

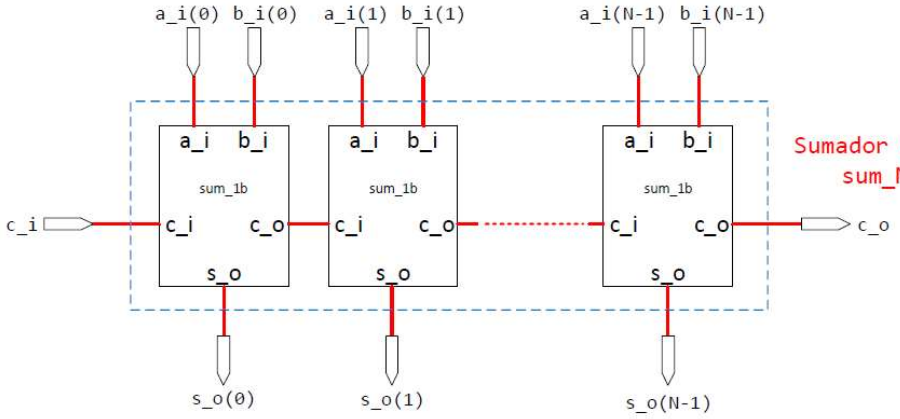




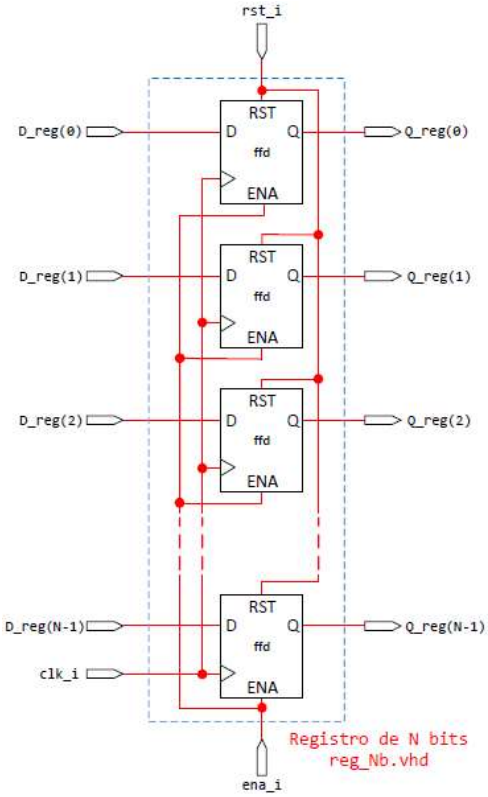
Sumador de 1 bit
sum_1b.vhd



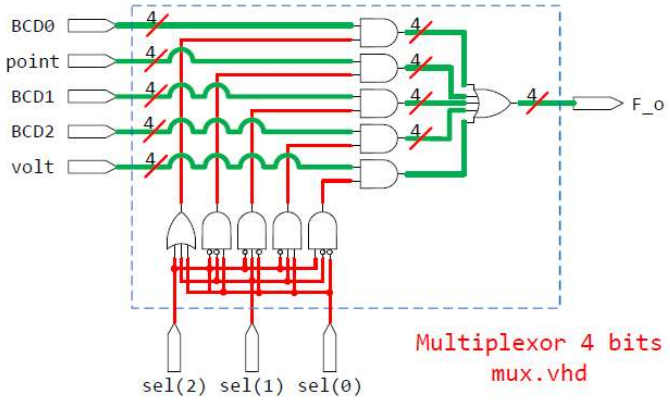
Comparador de N bits
comp_Nb.vhd



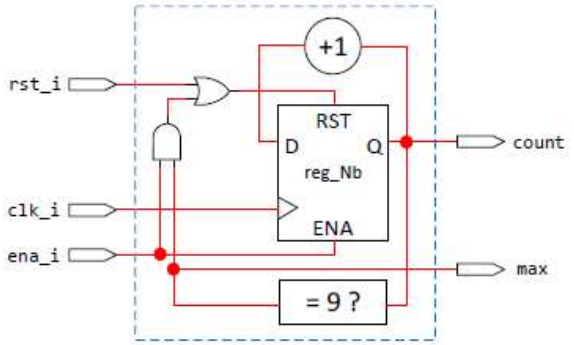
Sumador de N bits
sum_Nb.vhd



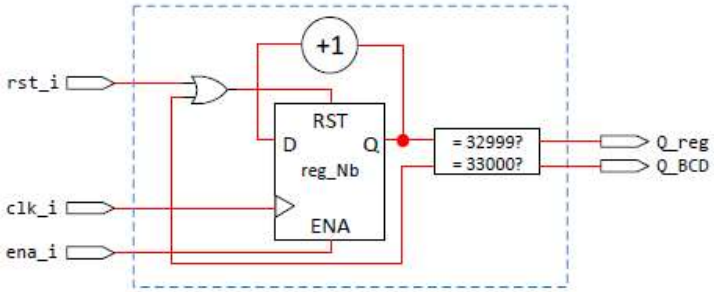
Registro de N bits
reg_Nb.vhd



Multiplexor 4 bits
mux.vhd

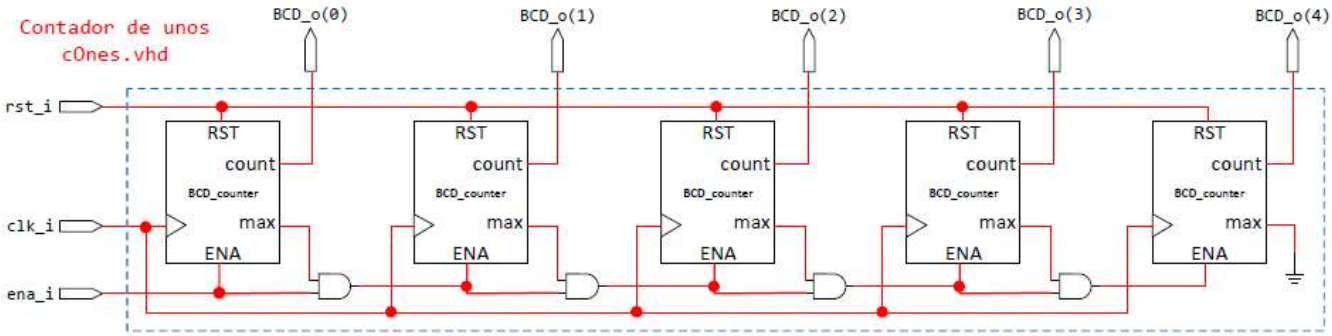


Contador BCD
BCD_counter.vhd

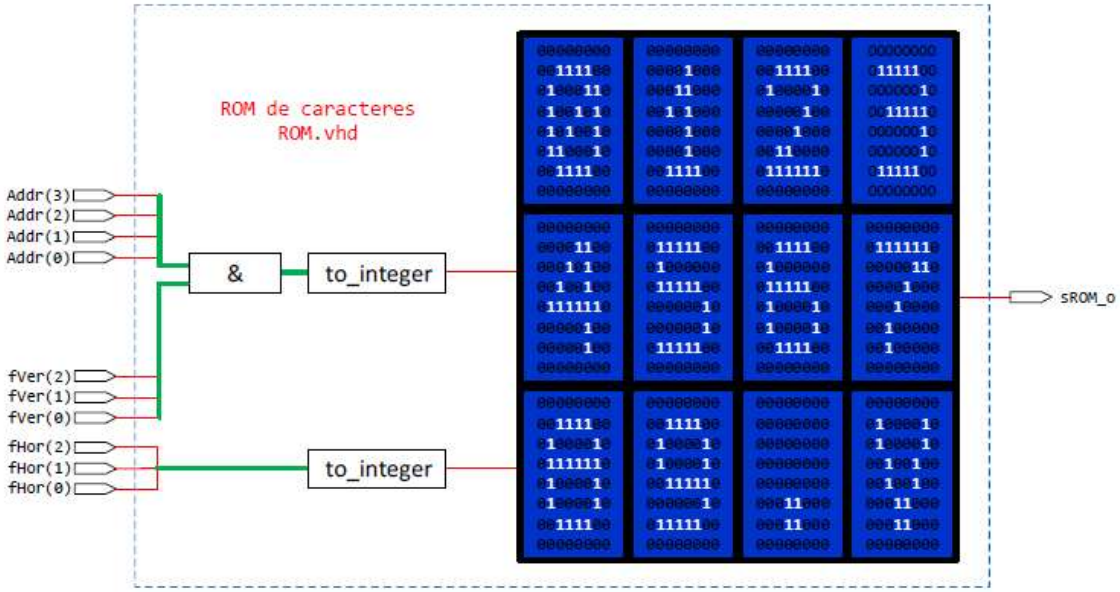


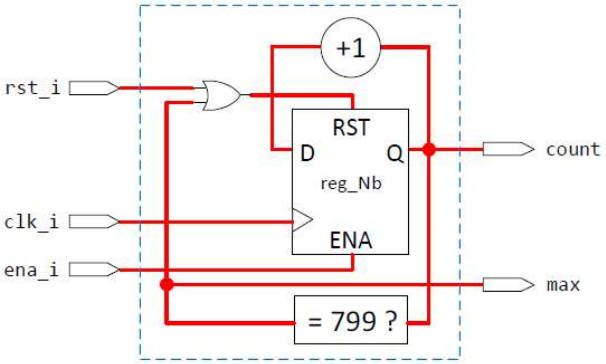
Contador binario
c33k.vhd

Contador de unos
cOnes.vhd

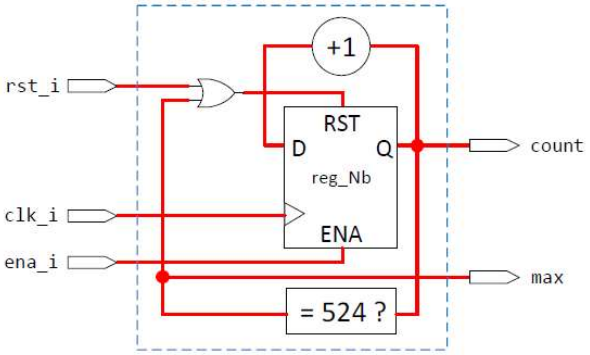


ROM de caracteres
ROM.vhd

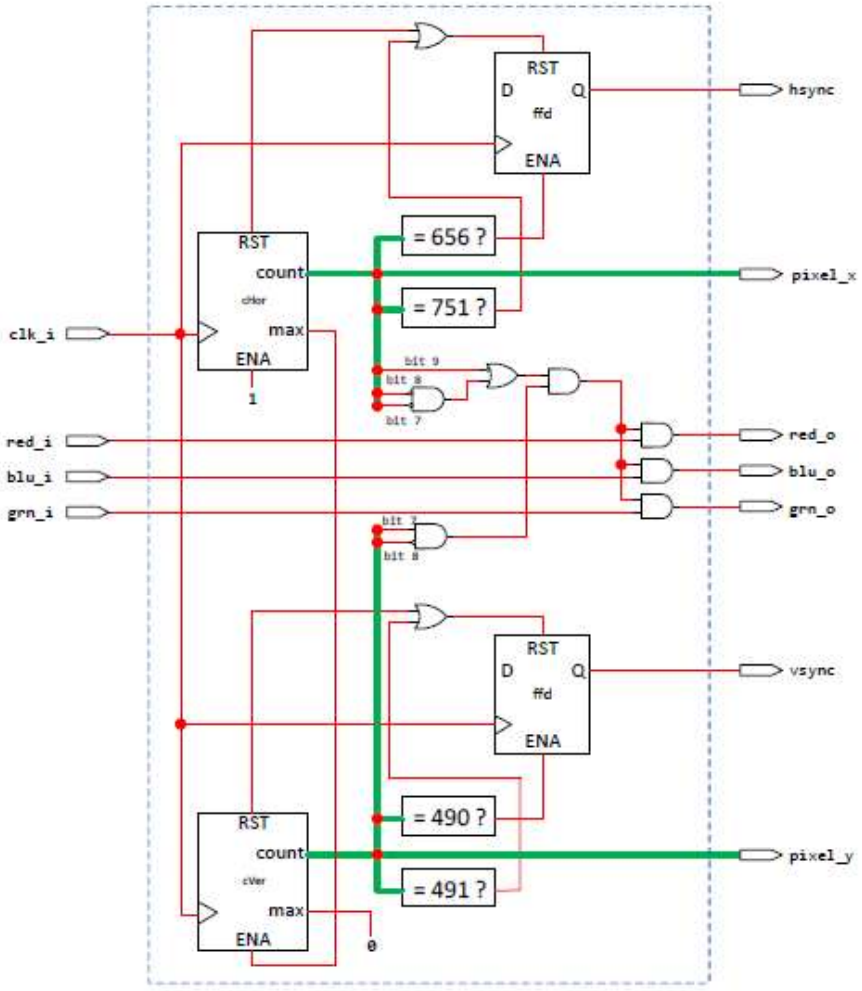




Contador Horizontal
cHor.vhd



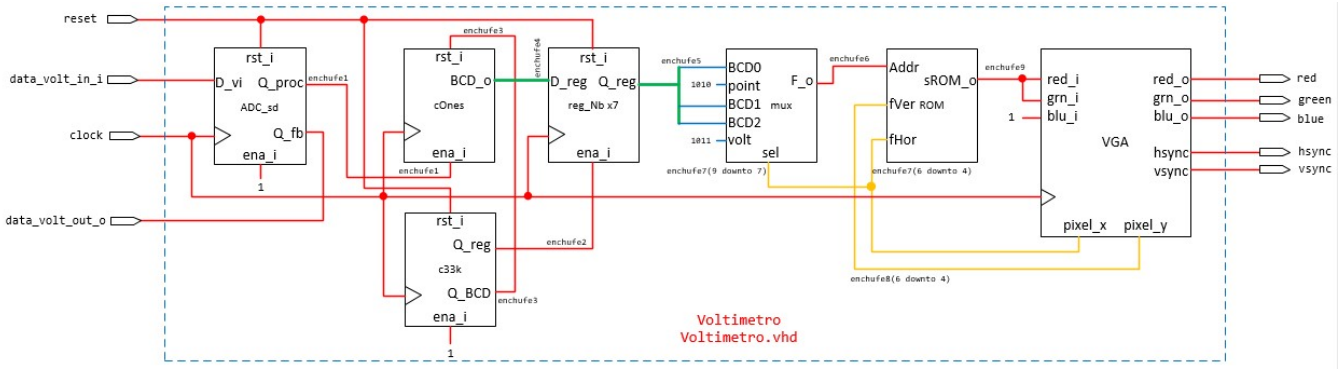
Contador Vertical
cVer.vhd



Video Graphic Adapter
VGA.vhd

En la VGA:

- Cables rojos: 1 bit
- Buses verdes: 10 bits



En este último, el voltímetro:

- Cables rojos: 1 bit
- Buses verdes: 7 x 4 bits
- Buses azules: 4 bits
- Buses amarillos: 3 bits