



UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Electrónica
Digital I

2018

Trabajo Práctico Final

Alumno

Calcagno, Misael Dominique

Legajo CYT-6322

Fecha de entrega: 14/02/2018



Objetivo

Desarrollar un voltímetro digital programado en lenguaje descriptor de hardware con salida VGA. Específicamente para una FPGA de Xilinx de la familia Spartan 3E modelo XC3S500E.

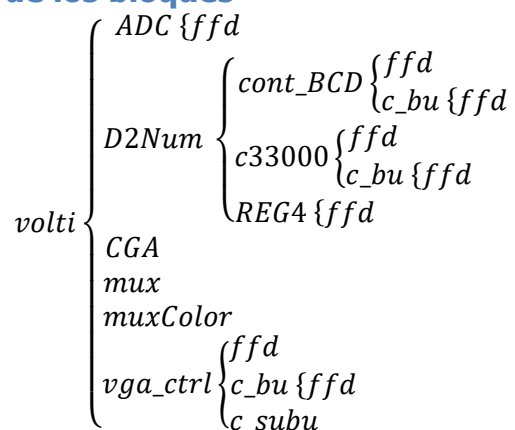
Resumen

Se presenta a continuación, el informe con los criterios empleados, esquemáticos, tablas de verdad y de Karnaugh por bloque y por último el reporte del programa ISE.

Consideraciones

Se debe únicamente utilizar lenguaje estructural para todo el trabajo, excepto por el Flip-Flop D que se usará behaviour para describirlo. Además, ha de ser lo más modular y eficiente posible. Las condiciones de impresión en pantalla son cinco caracteres, el dígito más significativo de la unidad, la coma, el primer decimal, el segundo y la "V" de volt. Se optó por adicionar una codificación de color dependiendo del nivel de voltaje que se detallará más adelante.

Jerarquía estructural de los bloques



Descripción de cada bloque

Board

En la Figura 1 se observa que se han dispuesto los switches de enable y reset a modo de controlar el voltímetro. Hay que tener en cuenta que, como el controlador de VGA no tiene entrada de reset, siempre se imprime en pantalla. Logrando una comunicación continua con el monitor sin pérdidas cuando se resetea.

volti

Como se observa en la Figura 2, el voltímetro cuenta con seis módulos. El ADC se encarga de generar los pulsos que utiliza para contar el D2Num. El multiplexor mux se encarga de seleccionar uno de los caracteres basándose en la posición horizontal en pantalla. El CGA determinando en qué pixel, o mejor dicho, bloque de ese caracter se encuentra imprimiendo decide si colorear blanco o negro. El muxColor, que no es realmente un multiplexor, determina el color de impresión basado en el nivel de voltaje. El vga_ctrl se encarga de coordinar la impresión con contadores verticales y horizontales para el sincronismo.



ADC

En la Figura 3 se presenta el ADC como un simple flip-flop D, con dos salidas. Una negada y la otra sin negar. Es Q_ADC la que otorga los pulsos para el D2Num. Mientras que vpositive y vnegative simplemente se encargan del aspecto analógico.

D2Num

Figura 4, los pulsos generados por el ADC entran vía O_ADC, que habilita a los cont_BCD para que cuenten en 33000 ciclos de clock. El c33000 se encarga de controlar esto, enviando la señal de reset a 33001 ciclos, pero avisando a los 33000 ciclos a los REG4 para guardar la información. Como el número total máximo que puede contar es 33000 que se corresponde con 3.30V, los últimos dos dígitos no se guardan. Por eso hay solo tres REG4.

cont_BCD

Figura 5, se trata de un contador binario que cuenta desde "0000" hasta "1001". Ya que la lógica hace que inmediatamente se alcance el "1010" se envía la señal de reset. Además, cuenta con un carry en "1001" para avisarle al cont_BCD siguiente que debe contar el siguiente pulso de clock.

c_bu

Figura 6, la compuerta XOR me garantiza que cuando el carry anterior y el número de la celda sean iguales, D valdrá '0' y Q valdrá eso en el siguiente ciclo. Esto es porque se debe generar, por ejemplo con dos bits, esta secuencia 00-01-10-11-00... Como se aprecia, el segundo bit cambia luego de dos ciclos de clock. Entonces, para extenderlo a n bits, solo se debe tener en cuenta los dos bits anteriores n-1 y n-2 para observar si son los dos '1' y los carries anteriores, entonces el carry va a ser '1' también y entra en la XOR del n bit.

C _{n-1}	n-1	C _n	n
0	0	0	0
1	1	0	0
0	0	0	1
1	1	1	1
0	0	0	0

c33000

Figura 7, un simple contador binario. Los recuadros "LOGIC" simplemente representan arrays de compuertas ANDs y ORs, nada que valga la pena resaltar ya que son funciones sencillas.

REG4

Figura 8, un conjunto en paralelo de cuatro flip-flops D para ser usados como registros.

CGA

Figura 9, basado en los registros de posición de pantalla y caracter se decide la posición dentro de la ROM cual es el bloque a imprimir.

mux

Figura 10, una serie de compuertas lógicas decide según una de las cinco franjas posibles en pantalla cual array de qué caracter imprimir en pantalla.



muxColor

Figura 11, no es realmente un multiplexor, ya que sus entradas a seleccionar serían fijas, siempre va a elegir algún color.

vga_ctrl

Figura 12, con dos contadores individuales se generan las posiciones horizontales y verticales. La razón porque el contador horizontal tiene un bit más es para bajar la frecuencia a la mitad. Como el contador vertical se habilita con el horizontal no hay necesidad de adicionarle un bit más. Los FLAGGER son simplemente flip-flops D que llegada a cierta posición horizontal o vertical, cambian de estado. Combinando FLAGGERS se obtiene la condición vidon que es la necesaria para determinar las únicas posiciones donde se ha de imprimir. Los SUBTRACTOR son arrays de celda sustractoras para tener la posición real en pantalla.

c_subu

Figura 13, basándose en una tabla de verdad se determinó esta celda unitaria de resta para un solo bit. Como la resta se sabe que va a ser de un número menor a uno mayor. No hace falta contemplar el caso en el que se agoten los carries. En la Figura 14 se presentan la tabla de verdad junto con los diagramas de Karnaugh y las funciones resultantes.

Reporte ISE

En la Figura 15 se tiene el reporte del ISE, donde se observan 32 warnings que corresponde a signals desconectadas. Se prefirió dejarlas desconectadas para mantener la legibilidad del código, porque eliminarlas, lo harían más complejo.

Figuras

En las siguientes páginas se presentan todas las figuras.