**BITACORA**

**LABORATORIO ELECTRONICA DIGITAL**

Nicolás Mesa Serna

Juan José Alba

**Jueves 15 de agosto. 2019.**

Se comenzó a investigar sobre las FPGA, sus usos o posibles aplicaciones y aclararon dudas que no se lograron resolver mediante la lectura al datasheet de la tarjeta con la ayuda del docente.

Acudimos a las siguientes fuentes:

<https://www.xilinx.com/support/documentation/data_sheets/ds312.pdf>

<https://reference.digilentinc.com/reference/programmable-logic/spartan-3e/reference-manual>

**Jueves 22 de agosto. 2019.**

El docente nos realizó una introducción al lenguaje de descripción de hardware Verilog, en la cual nos dios abrebocas a lo que próximamente seria nuestra próxima entrega, y nosotros complementamos viendo la documentación y algunos videos.

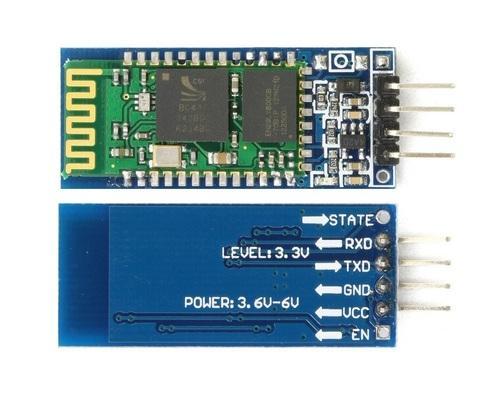
En el proceso de aprendizaje nos topamos con algunas dificultades, pero que prontamente pudimos solucionar gracias a la monitora y más documentación.

<http://www.iuma.ulpgc.es/~nunez/clases-FdC/verilog/Verilog%20Tutorial%20v1.pdf>

<https://www.youtube.com/results?search_query=verilog+tutorial>

**Viernes 23 de agosto. 2019.**

Nos dirigimos hacia el centro a realizar la compra del módulo bluetooth HC-06, chip que usaríamos en las siguientes entregas para la comunicación inalámbrica del dispositivo y la FPGA.



**Lunes 26 de agosto. 2019.**

Comenzamos a investigar sobre el protocolo RS232 y el UART, para lograr hacer la comunicación inalámbrica, fue un jordana para empaparnos de información, no hubo un avance visible, pero si logramos entender el funcionamiento en general.

Aafuentes:

<http://www.puntoflotante.net/RS485.htm#targetText=Esta%20norma%20fue%20dise%C3%B1ada%20para,velocidad%20m%C3%A1xima%20de%2019%2C200%20bps.>

<https://es.wikipedia.org/wiki/Universal_Asynchronous_Receiver-Transmitter#targetText=UART%2C%20son%20las%20siglas%20en,la%20tarjeta%20adaptadora%20del%20dispositivo.>

<http://www.electronoobs.com/eng_circuitos_tut26_code1.php>

<https://www.google.com/search?q=UART+UPITA&rlz=1C1CHBF_esCO799CO799&oq=uart+&aqs=chrome.0.69i59l2j0j69i60l3.1402j0j4&sourceid=chrome&ie=UTF-8>

**Martes 3 de Septiembre. 2019.**

Implementamos algunos códigos encontrados en diferentes páginas de internet, y en base a la funcionalidad de este se eligió el más apropiado pero tomando ideas de los otros complementamos nuestro conocimiento y el código.

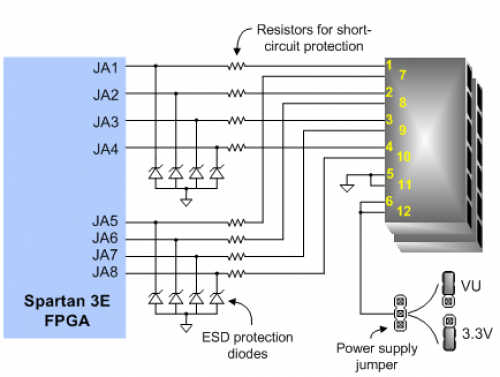
Acudimos a las siguientes fuentes:

<https://www.youtube.com/watch?v=2d6jz-DrXiU>

<https://www.youtube.com/results?search_query=uart+electronoobs>

**Martes 16 de septiembre. 2019.**

Hicimos el UCF con la ayuda de la monitora, buscando en el datasheet, los diferentes dispositivos de entrada y salida que se usarían para darle un sentido al módulo, no fue exitoso y no logramos encontrar dicho error ni con la ayuda de la tutora,



<https://khan.engr.uconn.edu/courses/ece4401_f14/Nexys2_500General.ucf>

**Miércoles 18 de agosto. 2019.**

Se corrigió el UCF, nos dimos cuenta que el error era un espacio entre el nombre de la señal y la posición del vector, cuando se concretó, conectamos la FPGA con el HC-06, se evidencio que el led del chip no estaba encendiendo, lo que significaba que no había paso de corriente, posteriormente notamos que el pin de tierra y voltaje era en la esquina.

Se impactó la tarjeta y por medio de una aplicación, logramos conectarnos por primera vez con dicho modulo, hubo señales de que un mensaje estaba llegando a la FPGA porque al enviar un carácter los leds pineados como salida del RX alumbraban, aunque no le encontrábamos mucho sentido al mensaje. Posteriormente nos dimos cuenta que la aplicación codificaba el carácter ASCII al número de índice, el cual se imprimía en binario por medio de los leds.



**Martes 23 de septiembre, 2019.**

El código sustentado al profesor fue el más elegante y mejor construido, pero tenía cierto error. Después de un exhaustivo análisis y revisión, se evidenció que la máquina de estados estaba bien diseñada, pero lo que le ocurría era que en las salidas pineadas (leds) no había rastro del dato enviado, es decir, no encendían los leds. A continuación adjunto algunas imágenes para ver el cambio y corrección del mismo:

**Antes:**



Acá podemos ver que se almacenaban los datos usando un índice.

**Después:**



Se implementa un registro de corrimiento.

**Miércoles 24 de septiembre, 2019.**

Pedimos prestada la FPGA para probar la solución propuesta por nosotros el día anterior, en ese instante notamos que los leds encendieron de manera correcta, por lo tanto pudimos evidenciar y establecer que la corrección era funcional.

Después de esto realizamos una investigación sobre lo que es la “Block Random Access Memory” (BRAM), para finalmente concluir que es memoria interna rápida y pequeña a la que se puede acceder en cada ciclo.

Acudimos a las siguientes fuentes

<https://forum.digilentinc.com/topic/5185-difference-between-bram-dram-and-dma/>

<https://www.youtube.com/watch?v=Ced8bhKTmek>