Manual de uso

Benjamín Opazo - Diego Lazcano

22 de Junio de 2018

En este documento se explicará como utilizar el diseño implementado. El contenido técnico del proyecto, así como el uso de los módulos en particular esta ampliamente detallado en el archivo *Informe Proyecto IPD432*. Este documento servirá como un recetario de uso, explicando los I/O, y como utilizar el archivo *test.m*.

Debe tomarse en cuenta que el diseño se implementó en una **Nexys 4 DDR**.

1 Diseño implementado

El diseño implementado está hecho de tal manera, que el usuario sólo deba cargar datos y hacer uso de un par de *switches*. El valor inicial de salida del hash es

$$6a09e667bb67ae853c6ef372a54ff53a510e527f9b05688c1f83d9ab5be0cd19$$

Este corresponde al valor *relajado* del módulo, que es cuando no ha hecho ningún cálculo. Se observa este valor porque el hash tiene valores iniciales que son comunes a todos los cálculos.

En la pantalla de 7 segmentos se ven los primeros 8 valores, es decir $6a09e667$. Apretando el botón *btnc* es posible avanzar por el hash de 8 en 8, es decir, al apretar el botón *btnc* se observa $bb67ae85$. Los led *LED15* a *LED13* corresponden al valor en binario de los 32 bytes que se avanzan, es decir, parten en 000, luego al apretar una vez el botón *btnc* son 001, y así.

El switch *sw[0]* se utiliza para levantar la bandera *driver_start*. Para eso es necesario tener precargados en RAM los datos necesarios para calcular el hash. Para cargar en RAM los datos diríjase a la sección **Uso de test.m**.

El *led[0]* corresponde a la bandera *sha256_hash_ok*, una vez que se apreta, se observará en la pantalla de 7 segmentos el valor inicial nuevamente.

Ahora se puede empezar de nuevo, levantando el switch *sw[0]* y calculando el hash para el segundo valor en memoria de la RAM.

2 Uso de test.m

El archivo *test.m* es a una implementación básica para cargar valores en la RAM del proyecto. Una vez que se tiene cargada la FPGA con el archivo .bit, se conecta al computador mediante puerto serial, y se cambia manualmente en la línea 8 del código el puerto serial que se quiere utilizar. Las variables *a1* a *a7* corresponden a los vectores de prueba que se cargarán en el computador. La siguiente tabla muestra los vectores de prueba del archivo, junto a sus respectivos hash sha256.

Vector	Valor Hexadecimal	Hash Sha256
a1	00 01 02 ... 1F 00 01 02 ... 1F	7c26fdd1387b71ca2b32c7778087933352bceddf1b767f4f9230aa543df6153a
a2	00 02 04 06 ... 1E	a395b57419a0e470d347e284e0d46c8974d306058a28d4d2bc7c1b3e06edee77
a3	FF FF ... FF (64 veces)	8667e718294e9e0df1d30600ba3eeb201f764aad2dad72748643e4a285e1d1f7
a4	01 09 00 07 01 09 09 04	463a52f2c4f9f0406b3fd7bd6bacf3369b9c2c6e41cecdfb1b7a4954365de0ed
a5	01 01 ... 01 (64 veces)	fb8e69bdfa2ad15be7cc8a346b74e773d059f96cfc92da89e631895422fe966a
a6	01	4bf5122f344554c53bde2ebb8cd2b7e3d1600ad631c385a5d7cce23c7785459a
a7	a1 (FF - a1)	486459d568e4b49f635763ef88ffe4c3639eb2393674ea300e351210612f1b23

Table 1: Valores que vienen en test.m y sus respectivos hash

Una vez que se corre el programa se cargan los datos en la RAM de datos junto a sus respectivos largos en la RAM de largos.

Nótese que los valores de *a1* a *a7* pueden cambiarse a gusto del usuario, y el script automáticamente calculará los largos y datos que debe enviarle a la FPGA.