

Resultados.

En este capítulo se presentan los resultados obtenidos en el proyecto, con la intención de validar la funcionalidad y también de encontrar los puntos fuertes y las falencias del mismo.

Configuración del Hardware

Las pruebas se desarrollaron en una placa de desarrollo Altera DE2 que cuenta con las siguientes características:

Feature	Description
FPGA	<ul style="list-style-type: none">▪ Cyclone II EP2C35F672C6 with EPCS16 16-Mbit serial configuration device.
I/O Interfaces	<ul style="list-style-type: none">▪ Built-in USB-Blaster for FPGA configuration▪ Line In/Out, Microphone In (24-bit Audio CODEC)▪ Video Out (VGA 10-bit DAC)▪ Video In (NTSC/PAL/Multi-format)▪ RS232▪ Infrared port▪ PS/2 mouse or keyboard port▪ 10/100 Ethernet▪ USB 2.0 (type A and type B)▪ Expansion headers (two 40-pin headers)

Memory	<ul style="list-style-type: none"> ▪ Built-in USB-Blaster for FPGA configuration ▪ Line In/Out, Microphone In (24-bit Audio CODEC)
Displays	<ul style="list-style-type: none"> ▪ Eight 7-segment displays ▪ 16 x 2 LCD display
Switches and LEDs	<ul style="list-style-type: none"> ▪ 18 toggle switches ▪ 18 red LEDs ▪ 9 green LEDs ▪ Four debounced pushbutton switches
Clocks	<ul style="list-style-type: none"> ▪ 50 MHz clock ▪ 27 MHz clock ▪ External SMA clock input

La FPGA incluida en la placa es una Cyclone II EP2C35 cuyas especificaciones son:

Feature	Description
LEs	33216
Total RAM bits	483840
Embedded multipliers	35
PLLs	4
Maximum user I/O pins	475

Parámetros

Tamaño de la FIFO

La FIFO es capaz de almacenar palabras de 72bits en una cantidad equivalente a 2^n siendo n el valor del parametro `AWIDTH`, que se encuentra especificado en `defines2.v`, en los resultados que se presentan en este capitulo n tendra el valor de 12, lo que arroja un tamaño total de 32KBytes(4096 palabras de 71bits), siendo esta la potencia de 2 mas grande que puede ser asignada en los bloques de memoria de la FPGA seleccionada.

Configuración del Generador

El generador entregara un flujo de paquetes uniforme, siendo la cantidad de ciclos de clock entre paquetes el valor utilizado para regular dicho flujo. Para esto solo basta modificar el valor superior de la variable Counter que se encuentra en el modulo `autoen.v`. La transformación entre la cantidad de paquetes por segundo y el valor que debe ser puesto en el modulo especificado es la siguiente:

$$Gap = (1 / (Paquetes por segundo * T_{clock})) - (Palabras por Paquete + 3)$$

Donde T_{clock} esta expresado en segundos, para un Clock de 50Mhz $T_{clock} = 2 \cdot 10^{-9}$ segundos, y Palabras por paquetes es la cantidad de palabras de 64bits por paquete. En esta implementación se configurara el generador para que entregue paquetes de 8 palabras de 64bits con la siguiente distribución:

1						64	
SOP (1 byte)		SA (6 bytes)				DA (byte 1)	
DA (bytes 2 to 6)				Length (2 bytes)		Ver (4 bit)	H. Len (4 bit)
Services (1 byte)	Total Length (2 bytes)	Identifier (2 bytes)	Flags (1 byte)	Offset (1 byte)	TTL (1 byte)		
Protocol (1 byte)	Checksum (2 bytes)	SA (4 bytes)				DA (byte 1)	
DA (bytes 2 to 4)							

donde el único valor que cambiará será el de IP Destino, ajustándolo mediante la variación del parámetro `IP_D_N` en el archivo `defines.v`, de acuerdo

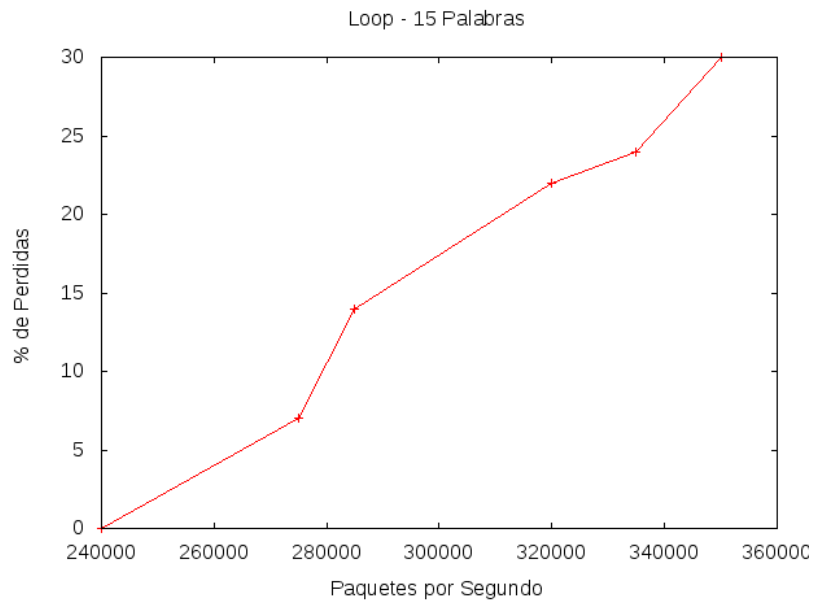
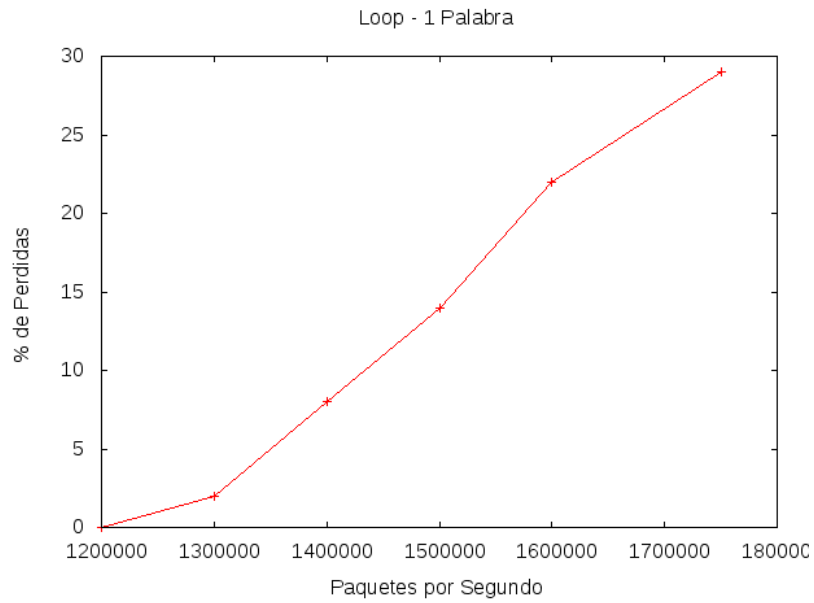
a la manera en la que se quiera ejercitar la tabla de lookup.

Porción del Cabecera enviada al procesador

Se contrastarán los resultados obtenidos mediante el uso de dos módulos Uplink diferentes. El primero le envía al Procesador las Primeras 5 palabras de 72bits, dado que el bus es de 32bits son necesarias 15 lecturas para poder completar la transferencia. Todo esto se hace en el transcurso de una única Interrupción. El segundo modulo solo envía al procesador el dato correspondiente al Campo IP Destino, como este valor tiene 32 bits una sola lectura es suficiente para su transmisión.

Caso loop

En primera instancia nos ocupamos de el caso vacío a los fines de encontrar los limites superiores de nuestro diseño. En este caso, el software solo se limita a recibir los datos e inmediatamente después confirma el procesamiento y envía los resultados de regreso al hardware. Se realizan las pruebas correspondientes para las dos versiones de Uplink.

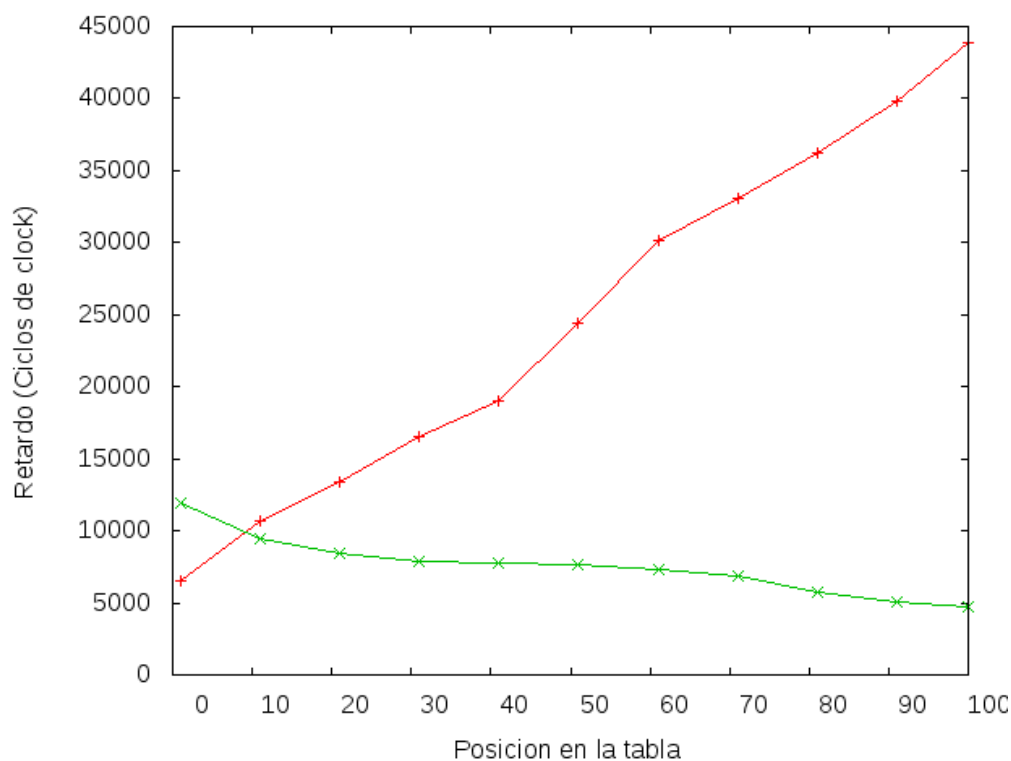


En el eje de las abscisas es posibles ver la cantidad de paquetes por segundo, el origen corresponde a la mayor velocidad a la que es posible transmitir sin perdidas. En las Ordenadas se puede observar la cantidad de paquetes perdidos en valores porcentuales, para obtener esta métrica se proceso una cantidad constante de paquetes, 9000, y luego se contrasto este valor con

un contador global que el Generador estampa en la ultima palabra de cada paquete. Así se calculo la cantidad paquetes perdidos, sobre la cantidad total de paquetes generados. Este mismo sistema es el usado en todos los graficos posteriores.

Algoritmos

Se estudiara la performance de los algoritmos aplicados, midiendo el retardo de lookup en funcion de la posición en la tabla de ruteo.



La curva roja corresponde a la búsqueda lineal y la verde, a la efectuada en el unibit trie.

Linear Lookup.

Como se ve en el gráfico, al efectuarse una búsqueda en forma iterativa nodo por nodo, el retardo es directamente proporcional a la posición del prefijo en la tabla, o dicho de otra manera, a la cantidad de iteraciones que se

deban realizar. El mejor caso es aquel en el cual se encuentre una coincidencia en el primer nodo de la lista y el peor es aquel en el cual se tenga que recorrer la lista entera para hallar el valor de decisión correspondiente.

Unibit Trie Lookup.

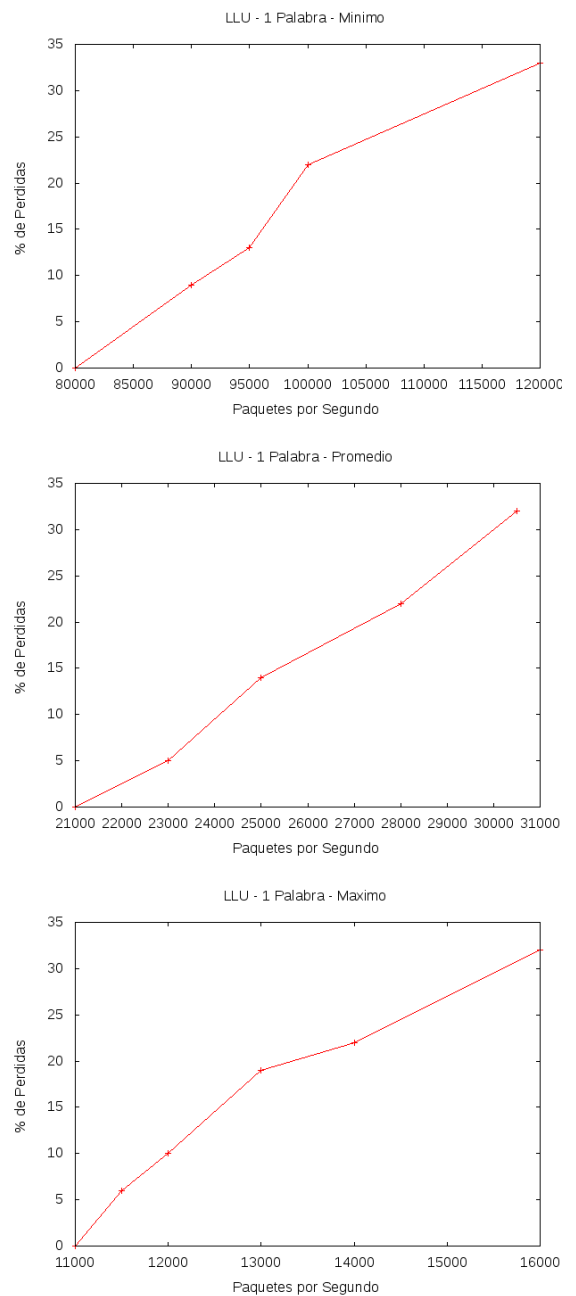
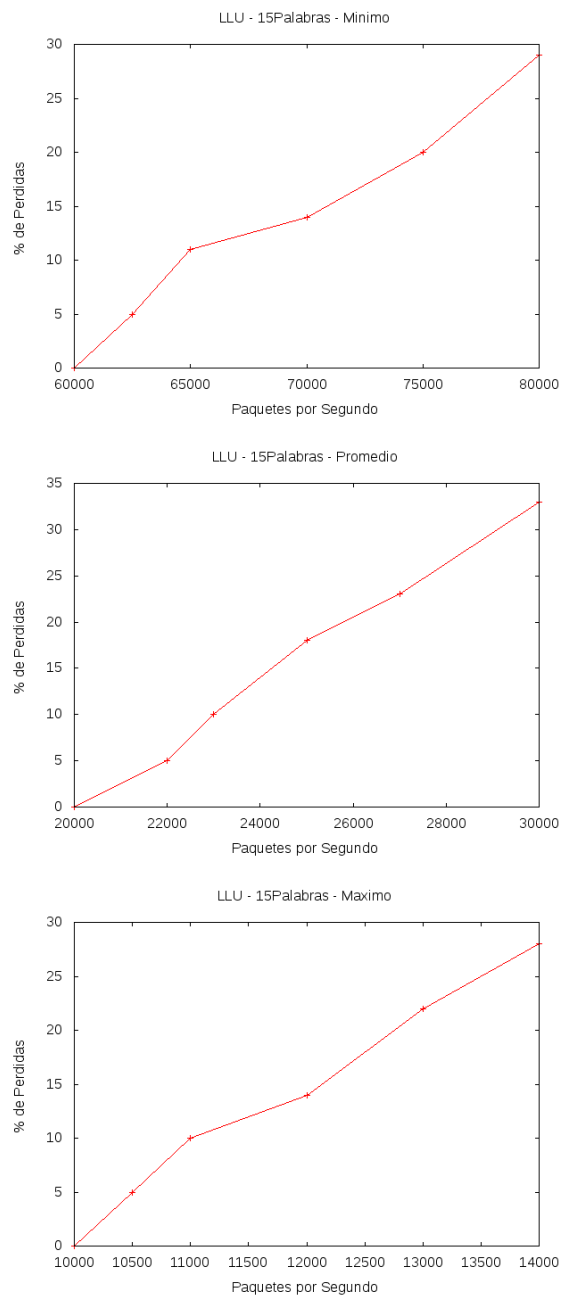
En el caso del árbol unibit, la búsqueda se efectúa en base a los bits que conforman el prefijo. En el peor de los casos la cantidad de iteraciones será igual a la longitud del prefijo más largo. En el mejor de los casos la decisión se encuentra en el nodo raíz. Por ello el tiempo de búsqueda será proporcional a la longitud del prefijo y no al tamaño de la tabla.

Vale hacer una aclaración sobre la curva del unibit trie. Para longitudes de prefijo menores a 6 el tiempo de búsqueda tiende a normalizarse. La explicación de este hecho yace en las características propias del algoritmo implementado. Si existe un nodo hijo asociado al valor del bit testeado, el puntero se desplazará hacia dicho nodo. Puede darse que no haya nodo hijo asociado para el valor del bit testeado y que el puntero haya quedado en un nodo que no contiene un valor de decisión. En tal caso, la función retorna el valor del último nodo decisión por el cual se haya pasado, y se habrán recorrido nodos innecesariamente. Ésto último hará que el retardo para longitudes de prefijos relativamente pequeñas tienda a homogeneizarse.

Completa

Se proba el funcionamiento completo del sistema. Para conseguir gráficos representativos se seleccionaran tres puntos en las curvas que indican los tiempos de accesos del algoritmos, un punto mínimo que corresponde al menor tiempo de acceso para el algoritmo en cuestión, un punto promedio que ejercita 10 entradas equidistantes a lo largo de la tabla y un punto máximo que indica el peor acceso posible.

LLU



Se puede observar que la diferencia entre la cantidad de paquetes que pueden ser transmitidos sin errores en el mejor caso(Mínimo) entre el modulo que envía una palabra al procesador y el que envía toda la cabecera, es con-

siderable, mientras que a medida de que se avanza en el recorrido de la tabla estos valores convergen, lo que era esperable ya que para accesos muy lentos a la tabla, el retardo introducido por el Hardware se vuelve despreciable.