Ciencia e Ingeniería
ISSN 1316-7081 *versión impresa*

Ciencia e Ingeniería v.28 n.1 Mérida ene. 2007

Como citar este artículo

Codificador de fuente programado en VHDL para dispositivos de hardware reconfigurable

Sandoval*, Cecilia y Felón, Antonio
Universidad de Carabobo, Dirección de Postgrado, Facultad de Ingeniería Naguanagua sector Bárbula Venezuela, 2008
Tel. (0241) 8672829 y 8674268 ext. 102, Fax: (0241) 8671655
* cecisandova@yahoo.com

Resumen
En este trabajo se presenta una metodología para el diseño modular de la etapa de codificación en VHDL, orienta la implementación sobre tecnología de matriz de compuertas programadas por campo (FPGA) y el estado del acerca de los métodos numéricos aplicados en el área de corrección de errores y procesamiento digital de seña por razones de desempeño y seguridad es preferible implementar los algoritmos en hardware. El desar metodológico se inicia con la definición de los componentes, y el modelo, descripción del comportamiento, lueg arquitectura es diseñada usando la sintaxis en VHDL y es capturado el diseño de hardware, finalmente se lleva a la validación de las salidas del diseño utilizando ModelSim 5.7 a través de simulaciones.
Palabras clave: VHDL, hardware re-configurable, codificadores, comunicación digital

Source coders programmed using VHDL for reconfigurable hardware devises

Abstract
In this paper we present a methodology to VHDL design of encoders and oriented to implementation with programmable gate arrays (FPGA) and the state-of-the-art about code, it is a basic for several algorithms in a error-correction codes and digital signal processing, due to performance and security reason, it is rather conver to implement algorithms by hardware. The methodology is initialized with the definition of the components and model, the description of the behavioral, later the architecture is designed and captures using syntax in VHDL, finally the simulations using ModelSim 5.7 and results allow validating the output of the design.
Key words: VHDL, hardware re-configurable, coders, communication digital.

Recibido: 11-01-2007 Revisado: 28-02-2007

1. Introducción

El algoritmo de Huffman es un algoritmo usado para compresión o encriptación de datos. Este algoritmo se bas asignar códigos de distinta longitud de bits a cada uno de los caracteres o símbolos que conforman un archi transmitir o almacenar. Si se asignan códigos más cortos a los caracteres que aparecen con mayor frecuenci consigue una compresión del archivo. Esta compresión es mayor cuando la variedad de caracteres diferentes aparecen es menor. Así mismo, para recuperar la información original es necesario conocer el código asignado a carácter, así como su longitud en bits, a fin de decodificar los datos recibidos, si ésta información se omite e trama de datos, y el receptor la conoce, podrá recuperar la información original. De este modo es posible utiliz algoritmo para encriptar archivos.

El diseño desarrollado cuenta con el establecimiento de un modelo matemático para desarrollar el códig programación de hardware, este fue optimizado para obtener alto desempeño, bajo costo y soluciones altam portátiles, a través de las características del lenguaje descriptor de hardware que ofrece alto nivel de paraleli para diseños específicos, Pérez (2002). La eficiencia de los diseños se obtuvo mediante el estudio de algorit originalmente planteados, manejo matricial a través de componentes y la aplicación de técnicas de programa modular. La metodología de programación sigue las normas de las fuentes investigadas IEEE (2002).

2. Bases teóricas

Este es un dispositivo ampliamente utilizado en el área de las comunicaciones digitales el cual convierte información de su formato original a una secuencia binaria conteniendo el menor número de bits posible, para lo se encarga de asignar de forma eficiente los bits para la representación de cada símbolo según la probabilida ocurrencia del símbolo, se le asigna menos bits a los símbolos de menor probabilidad de ocurrencia y mayor nur de bits a los símbolos de mayor probabilidad de ocurrencia [STREMLER 93] para así lograr la compresión de datos

de bits a los símbolos de mayor probabilidad de ocurrencia [STREHLER,95] para así lograr la compresión de datos

Entre los métodos de codificación de fuente se encuentra PCM, DPCM, codificación a través del modulador delta incluye varias de las etapas del sistema de comunicación.

2.1 Criterio de codificación de fuente óptimo

Al momento de diseñar un codificador de fuente se debe
tener presente el concepto de eficiencia del código, el cual esta ligado a la entropía (H), esta se define com menor número promedio de dígitos por símbolo; en la practica los codificadores requieren un número de dí promedio por símbolo mayor que la entropía, de manera que la eficiencia del codificador vendrá dada por la rela entre estos parámetros.

$$\eta_{CODE} = \frac{H}{\overline{N}} \qquad (1)$$

Para el cálculo de la entropía a la hora de diseñar el codificador se emplea

$$H = -\sum_{k=1}^{L} P_k \log_2(P_K) = -\frac{1}{\ln(2)} \sum_{k=1}^{L} P_k \ln(P_K) \qquad (2)$$

3. Metodologia

Para establecer el modelo matemático del codificador de fuente se estudio el codificador de Huffman y se estab la secuencia de pasos que componen la codificación para símbolos de longitud variable, como se descrit continuación.

3.1 Algoritmo de Huffman

Se realiza un conteo de las veces que aparece cada símbolo a codificar según la información que se vaya a compr y se crea una lista con los datos de símbolos y frecuencias.

Se ordena la lista de menor a mayor en función de la frecuencia de aparición de cada símbolo.

Se convierte cada elemento de la lista en un árbol.

Se fusionan todos estos árboles en uno único. Para esto mientras la lista de árboles contenga más de un eleme se sigue el siguiente proceso:

Con los dos primeros árboles formar un nuevo árbol, con cada uno de los árboles originales en una rama.

Sumar las frecuencias de cada rama en el nuevo elemento árbol.

Insertar el nuevo árbol en el lugar adecuado de la lista según la suma de frecuencias obtenida.

Para asignar el nuevo código binario de cada carácter sólo hay que seguir el camino adecuado a través del árbc se toma una rama cero, se añade un cero al código, si se toma una rama uno, se añade un uno.

Codifica la información según los nuevos códigos. A partir del primer paso se obtiene que la probabilidac ocurrencia del símbolo a es igual a 0,49 para los símbolos b y c la probabilidad es igual a 0,21 y para el símbolo probabilidad corresponde a 0,09. De esta información se construye el árbol que se observa en la Fig. 1.

Siguiendo el sentido de la codificación se establece la representación binaria de cada uno de los símbolos, com muestra en la Fig. 1.

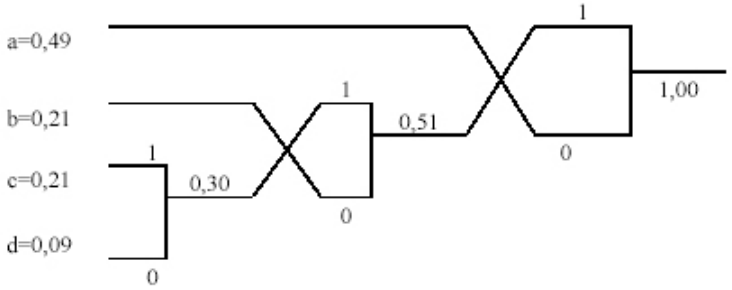

Fig. 1.

Tabla 1. Código fuente resultante

Symbols	Symbol probabilidad	Codeword Huffman	Prob. [Number of code Symbol]
a	0,49	0	0,49
b	0,21	10	0,42
c	0,21	111	0,63
d	0,09	110	0,27
			$\overline{N}$ = 1,81 dig/simb

A partir de las Ecs 2 y 3, se obtuvo la entropía y la eficiencia del código

$$H = 0{,}49\log_2\left(\frac{1}{0{,}49}\right) 0{,}21\log_2!\left(\frac{1}{0{,}21}\right) + 0{,}21\log_2\left(\frac{1}{0{,}21}\right) +$$

$$+ 0{,}09\log_2\left(\frac{1}{0{,}09}\right) = 1{,}762$$

$$\eta_{CODE} = \frac{1{,}762}{1{,}810} = 0{,}9738$$

Para una secuencia de entrada se realiza la traducción de acuerdo al código establecido, es decir; se codificar símbolos como se muestra en la Tabla 2.

Tabla 2. Asignación de códigos a los símbolos de entrada

A	B	A	A	D	A	A	C	A	B	A	C	D	A
0	10	0	0	110	0	0	111	0	10	0	111	110	0

El siguiente paso es empaquetar los bits en grupos de tamaño fijo para el procesamiento de los datos, para identi el inicio de cada símbolo se ha establecido un nivel de tensión débil que permita al decodificador reconstru información, si se desea encriptar los datos se puede eliminar el identificador de símbolo y programar una llave reconozca el tamaño de cada símbolo transmitido para realizar la reconstrucción sin que pueda ser decodificac información por ningún otro receptor. Lo que se muestra en la Tabla 3.

Tabla 3. Empaquetado de los simbolos codificados

0-0	-0-	111	-0-	0-1	10-	0-1	0-0	-11	0-1	11-
001	110	011	001	001	101					

Se observa la compresión, en la cual se transmite la información con un menor número de bits, si se transmiti los datos en código ASCII se requerirían 112 bits en tanto que en la data comprimida se transmiten 18 bits.

4. Análisis de resultados

Luego de describir el codificador bajo la sintaxis VHDL se obtiene para cada elemento su representación con có variable, los símbolos de longitud variable están representados a través de la señal /simb_cod esta salida se ge en paralelo a través de la entrada /fuente, el vector del símbolo de entrada se codifica y posteriorment empaquetan los bits resultantes, para el empaquetado se estableció una señal de reloj /clk_t a una alta frecue para reestructurar la trama de datos, la cual se obtiene como salida en la señal /out_paralelo , que presenta longitud de 3 bits, en una primera aproximación, se ha incluido un identificador de fin de símbolo, representadc un umbral de tensión que puede ser manejada por el FPGA para reconocer el fin de símbolo. Ver Fig. 2.

A través del código diseñado se logró codificar los datos provenientes de la fuente con una longitud variable permiten implementar tramas mas eficientes para la transmisión, este diseño corresponde a la primera etapa d sistema modular de comunicación y procesamiento de datos, lo cual optimiza la estructura de los mensajes pai codificación de canal, es decir; se puede agregar un cierto numero de bits, sin afectar significativamente el anch banda.

Finalmente, se programó el diseño empaquetando en /out _ paralelo la secuencia de bits codificados, en símbolo 3 bits, adicionando una señal /simb_long la cual especifica la longitud del símbolo codificado, con el propósit servir de referencia para la decodificación. Este es un diseño apropiado para criptografía en la cual el rece decodificará la data según una clave. Ver Fig. 3.

5. Conclusiones

El estudio de los casos del proceso de codificación permitió comprobar el correcto funcionamiento del diseño, obt su síntesis sobre un dispositivo FPGA, y procesar de forma paralela una trama de datos, con el propósito de dismi el consumo de recursos del FPGA se identificaron las operaciones para la implementación del algoritmo, e principios han sido útiles para el desarrollo de los diseños y comprensión de las bases de codificación y criptogr permitiendo una codificación con una eficiencia mayor a la original de la fuente.

Referencias

1. Ashenden PJ, 1990, The VHDL cookbook. First Edition. Dept. Computer Science, University of Adelaide, S Australia.

2. Disponible en http:// www.ashenden.com.au Carpio F, 1997, VHDL Lenguaje para descripción y modelad circuitos.

3. Chang KC, 1999, Digital systems design with VHDL and Synthesis, An integrated approach. IEEE Comp Society, USA.

4. Digilab, 2002, DIO2 Reference Manual. Disponible en www.digilentinc.com

5. IEEE Computer Society, 2002, IEEE Standard VHDL Language Reference Manual, The Institute of Electrical Electronics Engineers, Inc.

6. López FJ, 2005, Diseño de transmisor y receptor para redes inalámbricas W-MAN, Mención de Honor Málaga.

7. Nazar AS, 2004, Implementación eficiente de algoritmos criptográficos en dispositivos de hardware reconfigura Tesis Doctoral, México.

8. Pérez L, Serafín A y otros, 2002, Diseño de sistemas digitales con VHDL, Editorial Thomson, España. Disponibl http://www.dte.uvigo.es/vhdl/

9. Sandoval C, 2006, Transmisión de datos usando códigos Reed-Solomon e intercalado convolucional implemen sobre FPGA, Comunicación de Datos, Vol. 4, pp. 83-89, Colombia, Xilinx System Generator v2.1 for Simulink