JURNAL PUBLIKASI ILMU KOMPUTER DAN MULTIMEDIA
Halaman Jurnal: http://ejurnal.stie-trianandra.ac.id/index.php/jupkom
Halaman UTAMA Jurnal : http://ejurnal.stie-trianandra.ac.id/index.php

Penerapan Logika Algoritma Pada Prototipe Data Biner Untuk Generator Kartu FPGA

Indra Ava Dianta [a], Ahmad Ashifuddin Aqham[b]
[a] Fakultas Studi Akademik, indra @stekom.ac.id, Universitas Sains dan Teknologi Komputer Semarang
[b] Fakultas Studi Vokasi, ashif@stekom.ac.id , Universitas Sains dan Teknologi Komputer Semarang

ABSTRAK

The LFSR-based binary data sequence generator technique is used in various cryptographic applications and to design encoders / decoders in different communication channels. It is more important to test and verify by implementing on any hardware to get better effective results. Since FPGA is used to implement logical functions for faster prototype development, the existing LFSR design in the FPGA needs to be implemented to test and verify simulation and synthesis results between different lengths. The total number of random states generated in the LFSR depends on the feedback polynomial. The binary data generator is implemented using the LFSR shift register. This is a 23-bit shift register. The Random Number Generator allows you to generate random numbers of any length. The maximum length is 223−1

Keyword : *Binary data generator, LFSR, Implementation VHDL, Quartus FPGA*

Abstrak

Teknik generator urutan data biner berbasis LFSR digunakan berbagai aplikasi kriptografi dan untuk merancang encoder / decoder di saluran komunikasi yang berbeda. Lebih penting untuk menguji dan memverifikasi dengan mengimplementasikan pada perangkat keras apapun untuk mendapatkan hasil efektif yang lebih baik. Karena FPGA digunakan untuk mengimplementasikan fungsi logis untuk pengembangan prototipe yang lebih cepat, maka desain LFSR yang ada pada FPGA perlu diimplementasikan untuk menguji dan memverifikasi hasil simulasi dan sintesis antara panjang yang berbeda. Jumlah total status acak yang dihasilkan di LFSR bergantung pada polinomial umpan balik. Generator data biner diimplementasikan menggunakan register geser LFSR. Ini adalah register geser 23-bit. Random Number Generator memungkinkan Anda untuk menghasilkan nomor acak dengan panjang yang dipilih. Panjang maksimumnya adalah 223−1.

Kata kunci: Generator data biner · LFSR · Implementasi VHDL · Kuartus · FPGA

1. PENDAHULUAN

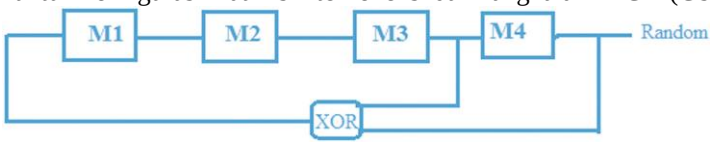
Untuk menghasilkan urutan data, bilangan acak sangat berguna dalam berbagai aplikasi seperti saluran komunikasi [1, 2]. Ini digunakan untuk merancang encoder dan decoder untuk mengirim dan menerima data dalam saluran komunikasi yang berisik. Mereka juga telah digunakan secara estetika, misalnya dalam sastra dan musik, dan tentu saja selalu populer untuk permainan [1]. Ketika kita berbicara tentang bilangan tunggal, bilangan acak adalah salah satu yang diturunkan dari sekumpulan nilai yang mungkin, yang masing-masing juga mungkin, yaitu distribusi yang seragam. Random Number Generator adalah perangkat komputasi yang dirancang untuk menghasilkan urutan angka. Ada berbagai metode untuk mengetahui bilangan pseudo-random. Kebanyakan dari mereka didasarkan pada persamaan linier dan membutuhkan sejumlah operasi aritmatika yang panjang. Di sisi lain, penggunaan register geser umpan balik memungkinkan pembuatan urutan biner yang sangat cepat. Urutan register panjang maksimum (urutan-m) sangat sesuai untuk mensimulasikan urutan biner yang benar-benar acak [3–5].

Konfigurasi FPGA biasanya ditentukan menggunakan bahasa deskripsi perangkat keras (HDL) yang mirip dengan yang digunakan untuk sirkuit terintegrasi khusus aplikasi (ASIC) (diagram sirkuit

Received Januari 30, 2022; Revised Maret 2, 2022; Accepted April 22, 2022

sebelumnya digunakan untuk menentukan konfigurasi, seperti halnya untuk ASIC, tetapi ini semakin jarang). FPGA dapat digunakan untuk mengimplementasikan fungsi logika apa pun yang dapat dilakukan oleh ASIC [6].

Aliran desain dari FPGA memungkinkan konfigurasi chip dari deskripsi awal dalam urutan langkah-langkah yang telah ditentukan sebelumnya yang dapat kita uraikan menjadi dua: desain dan simulasi. Tugas desain memungkinkan Anda beralih dari satu deskripsi ke deskripsi lainnya untuk sampai pada konfigurasi. Faktanya, sintesis logis memungkinkan untuk meneruskan dari deskripsi RTL arsitektur ke deskripsi di tingkat gerbang logis. Deskripsi elemen logis dioptimalkan sesuai dengan kecepatan, permukaan, atau batasan konsumsi yang diberlakukan oleh perancang. Alat sintesis menggantikan elemen logika umum dengan yang spesifik ke FPGA yang ditargetkan. Penempatan dan perutean mengubah deskripsi perangkat keras menjadi file konfigurasi. Alat sintesis menghasilkan file ini yang digunakan untuk konfigurasi matriks interkoneksi dari rangkaian FPGA (Gbr. 1).



Gambar 1. Diagram blok dasar LFSR

Kami menggunakan lingkungan QUARTUS II versi 7.0 dari perusahaan Alteras baik untuk desain maupun implantasi pada chip.

Quartus adalah perangkat lunak yang dikembangkan oleh perusahaan Altera, yang memungkinkan pengelolaan aliran desain FPGA secara lengkap. Perangkat lunak ini memungkinkan untuk membuat input grafis atau deskripsi HDL (VHDL atau Verilog) dari arsitektur digital, untuk mewujudkan simulasi, sintesis, dan implementasi target yang dapat diprogram ulang [10].

2. TINJAUAN PUSTAKA

2.1 Register Geser LFSR

LFSR adalah register geser yang bit masukannya merupakan fungsi linier dari keadaan sebelumnya. Fungsi linier yang paling umum digunakan dari bit sederhana adalah XOR. Jadi, LFSR paling sering adalah register geser yang bit inputnya dikontrol oleh eksklusif atau (XOR) bit tertentu dari nilai keseluruhan register geser [1, 8]. Nilai awal LFSR disebut seed. Karena register memiliki jumlah status yang mungkin terbatas, pada akhirnya register harus memasuki siklus berulang. Namun, LFSR dengan fungsi umpan balik dapat menghasilkan urutan bit yang tampak acak dan memiliki siklus yang sangat panjang. Aplikasi LFSR termasuk pembuatan bilangan pseudo-random, urutan derau semu, penghitung digital cepat, dan urutan pemutih. Implementasi perangkat keras dan perangkat lunak LFSR adalah Umum [7].

2.2 Implementasi PRNG Berbasis LFSR

Generator urutan nomor pseudo-acak dihasilkan dalam VHDL sesuai dengan rangkaian berikut sebagai fungsi dari konsep register geser. Bit dalam LFSR menyatakan bahwa mempengaruhi input disebut tap. LFSR panjang-maksimum menghasilkan sebuah urutan (apakah ia melintasi semua kemungkinan status $2n - 1$ dalam register geser kecuali keadaan di mana semua bit adalah nol), kecuali ia berisi semua nol, dalam hal ini ia tidak akan pernah berubah. Urutan angka yang dihasilkan dengan metode ini acak. Periode urutannya adalah $(2n - 1)$, di mana n adalah jumlah register geser yang digunakan dalam desain. Untuk menghasilkan register geser LFSR 23-bit, periodenya adalah 8388607. Ini cukup besar untuk sebagian besar aplikasi praktis. Susunan katup untuk umpan balik dalam LFSR dapat diekspresikan dalam aritmatika bidang finit sebagai mod polinomial. Ini berarti bahwa koefisien dari polinomial harus 1 atau 0. Ini disebut polinomial umpan balik atau polinomial karakteristik. Misalnya, jika katup berada pada 23, 18, polinomial umpan baliknya adalah

$$\mathbf{X23 + X18 = 1,}$$

2.3 Metode untuk Menghasilkan Urutan Data Biner

Generator Data diimplementasikan menggunakan LFSR, register geser yang bit masukannya merupakan fungsi linier dari keadaan sebelumnya. Fungsi linier yang paling umum digunakan dari bit sederhana adalah XOR. Jadi, LFSR paling sering adalah register geser yang bit inputnya digerakkan oleh eksklusif atau (XOR) bit tertentu dari nilai keseluruhan register geser.

3. METODOLOGI PENELITIAN

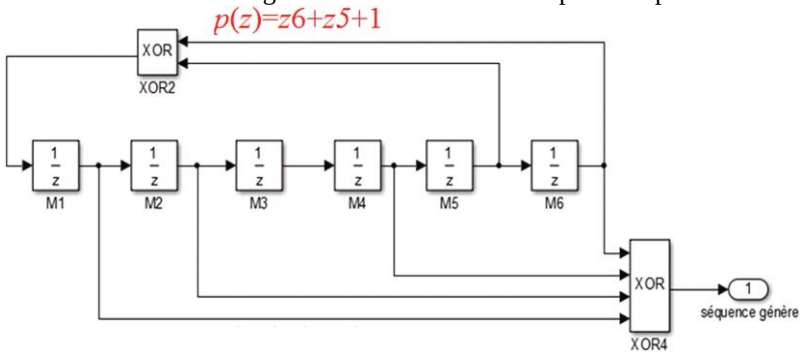
Generator Data diimplementasikan menggunakan LFSR, register geser yang bit masukannya merupakan fungsi linier dari keadaan sebelumnya. Fungsi linier yang paling umum digunakan dari bit sederhana adalah XOR. Jadi, LFSR paling sering adalah register geser yang bit inputnya digerakkan oleh eksklusif atau (XOR) bit tertentu dari nilai keseluruhan register geser.

3.1. Desain Generator Biner 6-Bit di Matlab

Untuk generator polinomial yang dipilih:

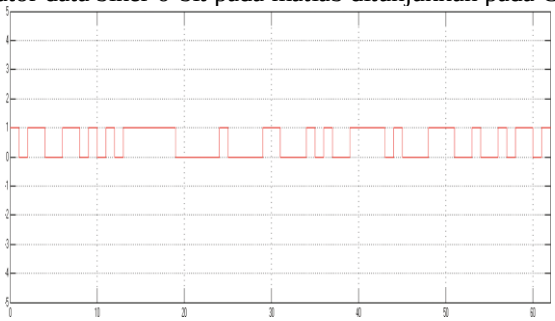
P (z) = z6 + z + 1,

Model Gambar 2 menghasilkan urutan data biner periodik periode 63.



Gambar 2. Diagram sirkuit LFSR 6-bit

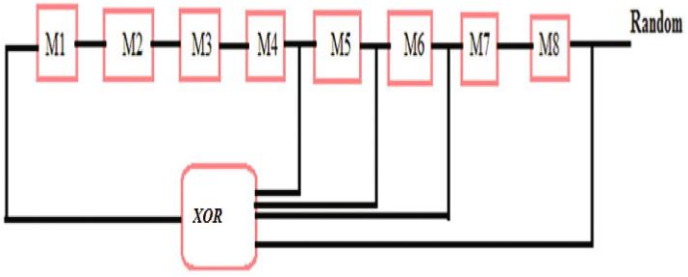
Hasil simulasi generator data biner 6-bit pada matlab ditunjukkan pada Gambar 3



Gambar 3. Generator simulasi 6 bit.

3.2. Desain Generator Data 8 Bit

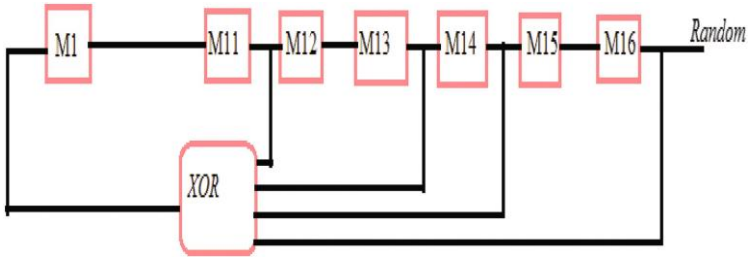
Generator 8-bit, dengan polinomial balik dengan panjang maksimum 8, menghasilkan 8 - 1 = 2 keluaran acak, yang diperiksa dari bentuk gelombang simulasi. Diagram rangkaian untuk LFSR 8-bit dengan polinomial panjang maksimum ditunjukkan pada Gambar. 4 [9].



Gambar 4. Diagram sirkuit LFSR 8-bit

3.3. Desain Desain Generator Data 16 Bit

LFSR 16 bit, dengan polinomial balik dengan panjang maksimum 3, menghasilkan 1 - 1 = 3 keluaran acak, yang diperiksa dari bentuk gelombang simulasi. Diagram rangkaian LFSR 16-bit dengan polinomial panjang maksimum ditunjukkan pada Gambar 5.

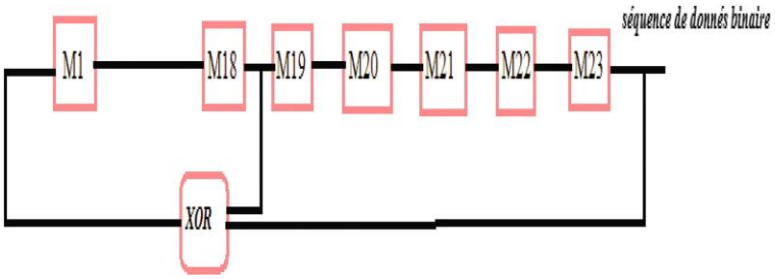


Gambar 5. Diagram sirkuit LFSR 16-bit

3.4. Arsitektur Generator Data Biner

Generator data biner direalisasikan menggunakan LFSR (Linear Feedback Shift Register). Gambar 6 menunjukkan struktur LFSR yang telah kami implementasikan. Ini adalah register geser 23-bit. Register secara rekursif didefinisikan menggunakan polinomial generator berikut:

$$X0 = X23 + X18$$



Gambar 6. Generator data biner

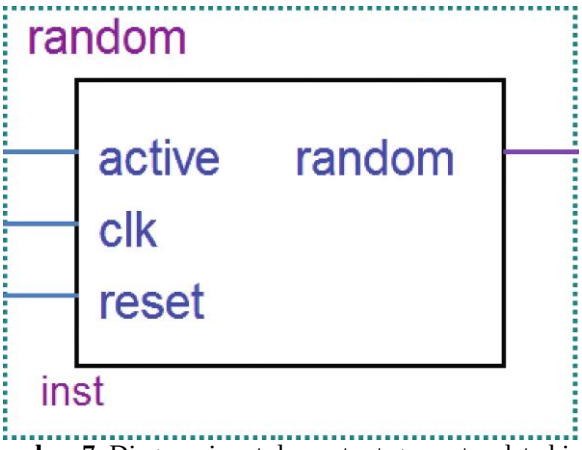
Memang, sinyal (x1, x2, dan x23) dari register memberikan nilai dalam representasi biner. Nilai register pada suatu periode yang bergantung pada polomial generator. Jadi LFSR tidak menghasilkan urutan data acak. Jika periode ini cukup besar, kami menganggap bahwa data yang dihasilkan acak. Struktur Gambar 6 memungkinkan untuk menghasilkan urutan data dengan periode 223−1.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Generator Data Biner

Generator data biner memiliki port input dan output yang ditunjukkan pada Gambar 7.

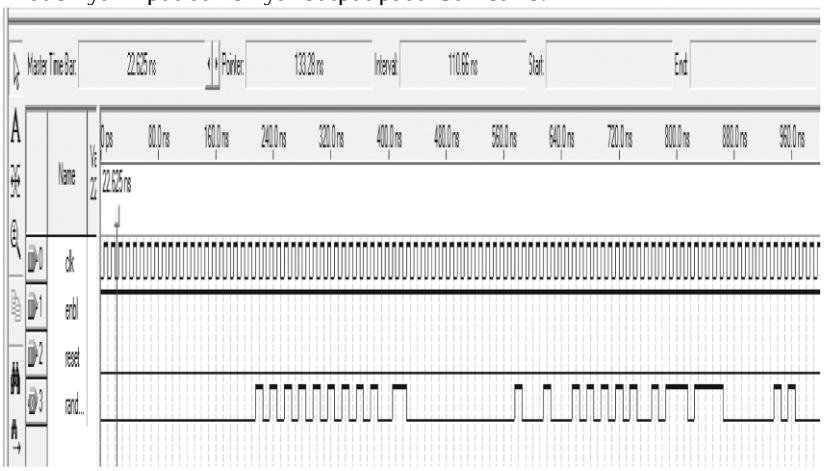
Clk : Sinyal jam aktif dalam status tinggi
Reset : Sinyal untuk mereset generator
Aktif : Sinyal untuk mengaktifkan generator
Acak : Sinyal keluaran dari bit yang dihasilkan oleh generator



Gambar 7. Diagram input dan output generator data biner

4.2. Simulasi Generator Data Biner

Untuk mencapai hasil simulasi, kami menguji generator data biner di lingkungan Quartus II, Anda dapat melihat sinyal input dan sinyal output pada Gambar 8.



Gambar 8. Simulasi generator

5. KESIMPULAN DAN SARAN

Dari penjelasan yang telah diuraikan pada bab – bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut :

Dalam rantai komunikasi digital, data dikodekan dan dikirimkan pada saluran komunikasi; sumber informasi adalah mata rantai pertama dalam rantai transmisi. Generator data biner memasok pesan yang membawa informasi digital.

Generator data biner menghasilkan urutan periodik yang tampak acak. Urutannya tidak acak secara statistik tetapi akan melewati banyak tes acak. Urutan ini disebut bilangan pseudo-random atau pseudo-noise.

Untuk perbaikan di masa mendatang, kami berharap dapat meningkatkan jumlah penyadapan, tingkat peluang serta panjang langkah dapat ditingkatkan dan sekali lagi dapat membandingkan urutan panjang maksimum antara LFSR dengan panjang yang berbeda dengan penerapan pada FPGA.

Dalam pekerjaan ini dan untuk tujuan perbaikan, pengontrol dapat ditambahkan ke generator data biner untuk membagi frame yang memasuki encoder untuk menambahkan bit redundansi dan dikirim melalui saluran transmisi.

DAFTAR PUSTAKA

[1] Bhasker, J.: A VHDL Primer. P T R Prentice Hall, Englewood Cliff s (2013)

[2] Brown, S., Vranesic, Z.: Fundamental of Digital Logic Design with VHDL, 2nd edn. McGraw Hill

[3] Goresky, M., Klapper, A.M.: Fibonacci and Galois representations of feedback-with-carry shift registers. IEEE Trans. Inf. Theory 48, 2826–2836 (2002)

[4] Hao, J., Li, Z,: On the production of pseudo random Numbers in Cryptogrzphy. J. Chzngehou Teach. Coll. Technol. 7 (2001)

[5] Hao, J., Li, Z.: FPGA design flow based on a variety of EDA tools. Micro-comput. Inf. 11-2(23), 201–203 (2007)

[6] http://www.polytech.univmontp2.fr/pravo/cours/Logique/Altera/Notice%20Quartus.pdf

[7] Katti, R.S. Srinivasan, S.K.: *EffiCient Hardware Implementation Of A New Pseudo-Random Bit Sequence Generator*. In: IEEE International Symposium on Circuits and Systems, ISCAS 2009 (2009)

[8] L'Ecuyer, P.: *Random Numbers For Simulation*. Commun. ACM 33, 10 (1990)

[9] Luby, M.: *Pseudo Randomness And Cryptographic Application*s. Princeton University Press, Princeton (1996)

[10] Panda Amit, K., Rajput, P., Shukla, B.: Design of multi bit LFSR PNRG and performance comparison on FPGA usingVHDL‖. Int. J. Adv. Eng. Technol. (IJAET) 3(1), 566–571 (2012)