Applying M-sequences Decimation to Generate Interleaved Sequence

Dang Van Truong, Le Chi Quynh

Abstract—M-sequences are widely used in for many purposes, from synchronization, whitening, communications and cryptography. We analyze decimation techniques and introduce two methods to generate decimation sequences which don’t have to calculate intermediate states. Then we apply these methods to interleaved sequence as a new method to pre-calculate for set of interleaved order which is more effective in implementation.

Tóm tắt—M-daisy được sử dụng rất rộng rãi trong nhiều lĩnh vực, từ việc đồng bộ, làm trắng thông tin, viễn thông và kỹ thuật mật mã. Chúng tôi phân tích kỹ thuật phân rã m-daisy theo bước và giới thiệu hai phương pháp sinh dãy phân rã theo bước mà không cần tính các trạng thái trung gian. Áp dụng phương pháp này vào daisy lồng ghép, ta có một phương pháp mới để tính trước tập các thứ tự lồng ghép có tính hiệu quả trong cải đạt thực tế.

Keywords—decimation, fast decimation, Fibonacci method, interleaved sequence, m-sequence, pseudo-random number generation.

Từ khóa—daisy lồng ghép, m-daisy, phân rã theo bước, phân rã nhanh, phương pháp Fibonacci, sinh daisy giả ngẫu nhiên.

I. INTRODUCTION

M-sequence are widely used in everyday technology. Many methods have been developed to construct pseudo-random sequences with better properties by transforming from m-sequences. One of the earliest of which is the decimation method. We can apply decimation method to generate pseudo-random sequence directly, or we can use this method to improve generating process of interleaved sequence.

When generate decimation sequences directly, we need to calculate many unnecessary internal states. In this contribution, we introduce two method to compose decimation sequences without calculation of useless states using pre-calculated matrix.

With interleaved sequence [4, 5, 10], we can apply decimation calculus as a new method to determine interleaved order set. We treat the sub-sequences of interleaved sequence as decimation sequences from origin m-sequence. Then we use the new decimation methods to get the first part of interleaved order set.

II. DECIMATION TECHNIQUE

A. Theoretical background

Let A be an m-sequence with periodicity $2^n - 1$ and root $\alpha$. We construct new sequence $A(T)$ by sampling every $7^th$ bit of A, starting with the first bit of A. $A(T)$ is decimation sequence called decimation of order $T$ from A, with $T$ is the decimation step.

It is known that if the decimation step $T$ and the periodicity $2^n - 1$ are co-prime, $(T, 2^n - 1) = 1$, $A(T)$ is also an m-sequence with the root $\alpha^T$, so-called m-sequence with difference generator polynomial but the same periodicity.

For $T$ equals $2^n$, $A(T)$ is a shifted version of A. In case $T$ and $2^n - 1$ are not co-prime, the generated decimation sequence is a LFSR sequence with periodicity [6]

$$\frac{2^n - 1}{\gcd(2^n - 1, T)}$$

(1)

The generator polynomial of this sequence is irreducible, but it is not primitive polynomial in almost case.
If decimation steps and cycle length satisfied interleaved sequence condition [5], then the new generator polynomial is primitive, but new polynomial degree is a divisor of primary degree.

In general application of m-sequences decimation method, step and cycle length must be co-prime in order to have decimation sequence with maximum cycle length and good random properties.

When apply decimation technique on m-sequence, we can change polynomial without changing in algorithm implementation. Decimation progress can be implemented in software only with appreciate calculation, no need to change in hardware design, make it is easier in implementation on small devices.

**Demonstration of an m-sequences decimations**

Assume that we have an m-sequence with order \( n = 23 \) and the generator polynomial:

\[
f = x^{23} + x^{18} + x^{15} + x^{14} + x^{11} + x^9 + x^5 + x^2 + 1
\]

with initial state \( S(0) = (1, 0, 0, \ldots, 0) \), or \((00000 \ 1)\) in hexadecimal.

Apply m-sequence generation formula, we can have next 18 internal states of m-sequence below (display in hexadecimal):

\[
\{ 400000, 200000, 500000, 280000, 548000, 6A0000, 750000, 3A8000, 5D4000, 2EA000, 175000, 4BA800, 25D400, 12EA00, 097500, 44BA80, 625D40, 312EA0 \}
\]

We show decimation of this m-sequence with decimation step \( T=3 \) and \( T=5 \) in the figure below.

**Fig. 1.** Decimation of m-sequence with decimation step \( T=3 \) and \( T=5 \).

We create small software to simulate decimation operation. In this software, we use Belekamp-Massey algorithm to determine the generator polynomial of newly generated decimation sequence. Running the software with different polynomials and decimation steps, we have the following result in Table I.

**TABLE I. RESULT OF ANALYSE DECIMATION SEQUENCES BY APPLICATION**

<table>
<thead>
<tr>
<th>No</th>
<th>Deg</th>
<th>Polynomial</th>
<th>Step</th>
<th>New Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24</td>
<td>( x^{24} + x^{20} + x^{19} + x^{17} + x^{15} + x^{12} + x^{10} + x^9 + x^8 + 1 )</td>
<td>11</td>
<td>( x^{23} + x^{22} + x^{20} + x^{17} + x^{14} + x^{11} + x^9 + x^5 + x^3 + x^2 + 1 )</td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>( x^{23} + x^{18} + x^{15} + x^{14} + x^{11} + x^9 + x^3 + x^2 + 1 )</td>
<td>5</td>
<td>( x^{23} + x^{22} + x^{18} + x^{17} + x^{16} + x^{15} + x^{12} + x^{10} + x^7 + x^6 + x^3 + x^2 + x + 1 )</td>
</tr>
</tbody>
</table>
Using this method, we don’t have to save T-1 intermediate internal state, but when calculating polynomial remainder by g(d) we still have to calculate remainder for T coefficient. This computation take some works, especially when T grow large.

When generate sequence by Fibonacci method, which is common way to generate sequence in micro controller

We store internal state of m-sequence in a shift register denote by ai (i=0..n-1)

To operate an m-sequence, each step require calculation of “feedback” bit an:

\[ a_n = \sum_{i=0}^{n-1} a_i \cdot f_{n-i} \]  

Then we shift all bits of the shift register to the right position, and put newly calculated bit an in bit n-1. We can take the discarded bit a0 as output bit of this step.

Apply decimation method with Fibonacci method, we have to calculate (4) for T times to generate one output bit. We can perform decimation method faster by pre-calculate in such way:

For example, we re-use m-sequence degree n = 23 with generator polynomial

\[ f = x^{23} + x^{18} + x^{15} + x^{14} + x^{11} + x^9 + x^5 + x^2 + 1 \]

The decimation step is T = 5

Analyze by application we found the new generated polynomial:

\[ g = x^{23} + x^{22} + x^{18} + x^{17} + x^{16} + x^{15} + x^{12} + x^{10} + x^7 + x^6 + x^5 + x^2 + x + 1 \]

From the initial state \( S_0 = \{a_0, a_1, ..., a_{23}\} \), we can build formula for next 5 bits:

\[ a_{23} = a_0 \wedge a_5 \wedge a_8 \wedge a_9 \wedge a_{12} \wedge a_{14} \wedge a_{18} \wedge a_{21} \]

\[ a_{24} = a_1 \wedge a_6 \wedge a_9 \wedge a_{10} \wedge a_{13} \wedge a_{15} \wedge a_{19} \wedge a_{22} \]
We have to shift the internal register by $T$ position onetime only. In traditional method, we have to shift the internal register $T$ times, each time it shifts one position. When using micro controller, this operation takes time to move data in memory. In hardware implementation, it takes $T$ clocks to complete this operation.

IV. APPLYING DECIMATION METHOD TO INTERLEAVED SEQUENCE

Interleaved sequence [5] is construct from m-sequence with degree $n$, which $n = m.l$

Choose the interleaved step $T$ satisfied:

$$T = \frac{N}{L} \text{ where } L = p^{m}.1, N = p^n.1$$

Interleaved sequence is linked from sub-sequences, the $k^{th}$ sub-sequence is a decimation of order $T$ from the origin m-sequence shifted by $k$ step. As show in [4], each sub-sequence is shifted version of full cycle m-sequence with degree $m$.

To generate interleaved sequence we need to determine the phase shift of each sub-sequence, called interleaved order set. There are 3 methods to pre-calculate the interleaved order set [5], using d-Transform, trace function and direct computation.

We can apply decimation generation methods above to get the first sub-sequence, which is the decimation of order $T$ from the origin m-sequence. But we know than this sub-sequence is a full cycle m-sequence with degree $m$, so we only need to compute the first $m$ bit of this sub-sequence. Using (4) with the generator polynomial of sub-sequence and the degree change to $m$, we have the remaining bits.

When we compute the first $m$ bit of the sub-sequence, we also have $m$ internal states of origin m-sequence at position $0, T, 2T \ldots (m-1)T$. From these states, using (4) we can determine the first $m$-bit of the second sub-sequence, and so on…
By using this method, we can calculate all the initial states of all sub-sequences. From these initial states, we can build interleaved order set of interleaved sequence. In memory usage, we need memory space to store \( m \) states of origin \( m \)-sequence with memory size \( m.n \).

In practice, we often generate the first part of interleaved sequence for a given length. In this case, we don’t have to determine all the interleaved order set. We just calculate which initial states as needed to generate such given length output.

CONCLUSION

From definition of decimation sequence, we introduce two method to speed up generating decimation sequence, even with a large step: using \( d \)-Transform and using pre-calculated transition matrix. The matrix method has the advantage of register shifting, so it is more effective in practice.

Applying these methods to interleaved sequence, we can directly determine the beginning part interleaved order set, with a little memory store needed. Using the introduced methods, we can effectively generate interleaved sequence for a given length, especially when the origin degree is large.

There is still another problem to be concerned, when the value of \( T \) is very large, these methods still need to be improved.

REFERENCES


ABOUT THE AUTHORS

Dang Van Truong
Workplace: Institute of Science and Technology of Cryptography
Email: truongdv@gmail.com
Education: He received a bachelor's degree in cryptography technology in 1996, received a master's degree in computer science in 2003 from the Le Quy Don University, doctoral student at Posts and Telecommunications Institute of Technology from 2014.
Recent research direction: Computer science, Information security.

Le Chi Quynh
Workplace: Electric Power University
Email: quynh.lechi@gmail.com
Education: He received a doctor's degree in Electronics in 1987.
Recent research direction: Number theory, Electronic and Communication, Information security.