Implementation of Golay Complementary Code Sequences Generator Based on FPGA

Dhamyaa H. Mohammed
Computer& Communication Engineering, Al-Mansour University College/ Baghdad
Email:dhamyaa.Husam@yahoo.com

Received on: 18/10/2012 & Accepted on: 4/4/2013

ABSTRACT

Golay sequences have some properties make it distinctive in the applications and results. However, for this distinction must select the code sequences carefully and accurately. Therefore, to satisfy these requirements, a creation algorithm must be easy, accurate and powerful. In this paper, an FPGA based, design and implementation of Golay complementary code sequence(GCCS) creation and then made autocorrelation between their pair codes to verify properties. The process time for proposed algorithm is less than that for all possible algorithm by (1/4 to 1/1024 for 4-bit to 16 bits respectively). Thus, the Search can be regarded as pioneers of the research application of this technique to the subject and got good results. The Implementation was based on 8-bit pair code and made by Xilinx-spartan-3A XC3S700AFPGA, with 50 MHz internal clock.

Keywords: Golay complementary sequences, FPGA-Spartan and correlator.

تنفيذ لمولد متسلسلة غولي المرمزة المكملة مبني بمصفوفة بوابات المجال القابلة للبرمجة موقعيا

الخلاصة

لتسلسل غولي بعض الخصائص التي تجعلها متميزة في مجال التطبيقات و النتائج. لذا لا بد لهذا التمييز من اختيار الرمز المتسلسل بعناية و دقه. ولتحقيق هذه المتطلبات, الخوارزمية المعدة يجب ان تكون سهله ودقيقه وفعاله. في هذا البحث، واعتمادا على مصفوفة بوابات المجال القابلة للبرمجة موقعيا تم تصميم وتنفيذ عمل متسلسلات غولي الزوجية التكميلية (او التكاملية) ثم عمل العلاقات التبادلية للتحقق من خصائصه. الوقت المستغرق لعمليات الخوارزمية المقترحة اقل من العمل على كل الاحتمالات بمقدار ٤/١ الى ١٠٢٤/١ الممثلة ب ٤ بت الى ١٦ بت على التوالي. يمكن اعتبار البحث من طلائع البحوث بتطبيق هذه التقنية على الموضوع وحصلنا على نتائج جيدة. تم تنفيذ بواسطة Xilinx-spartan-3A XC3S700A FPGA بتردد نبضي داخلي بمقدار ٥٠ ميكا هيرتز.

INTRODUCTION

https://doi.org/10.30684/etj.31.11A11
2412-0758/University of Technology-Iraq, Baghdad, Iraq

Golay sequences have some properties that make it distinctive in the applications and results. However, for this distinction, the code sequences must be select carefully and accurately. The property of Golay complementary sequences can be expressed mathematically, where a_i and b_i (i = 1,2, . . ., n) are the pair of binary complementary sequences of code length 2^n. The Aperiodic Auto-correlation Functions (AACFs) for the complementary sequences can be expressed as follows [1]:

$$c_j = \sum_{i=1}^{n-j} a_i \, a_{i+j} \qquad \ldots (1)$$

and

$$d_j = \sum_{i=1}^{n-j} b_i \, b_{i+j} \qquad \ldots (2)$$

The sum of the pair of AACFs can be expressed as;

$c_j + d_j = 0 \quad j \neq 0$ for specific unique pair and there relates as in futures point 7

$c_j + d_j \neq 0 \quad j \neq 0$ for other …(3)

and

$$c_0 + d_0 = 2n \qquad \ldots (4)$$

There are several recursive and one non-recursive methods for generating complementary sequences. The recursive method is used to generate the binary complementary sequences, which based on the following algorithm [1];

$$a_n = [a_{n-1}, b_{n-1}] \quad , \quad b_n = [a_{n-1}, -b_{n-1}] \ldots (5)$$

Where the operator [ ] denotes concatenation of sequences, and a_n and b_n represent the complementary binary sequence of length 2^n. If DFT taken the above equation, then get [2].

$$|c(k)|^2 + |d(k)|^2 = 2Nc \qquad \ldots (6)$$

There are many studies for generation a Golay complementary code sequences (GCCS). These are;
Jimenez and etal 2002[3], built the GCCS simulation by Monte-Carlo algorithm, and then by DSP TMS320C6201. Borislav and etal 2004[4], present a computer algorithm to generate a GCCS for modern application. Carlos and Changho 2007[5,6], a recursive algorithm for reduce the computational of GCCS creation are present. Jonathan 2009 [7], generates GCCS from Barker code.

PROPOSED CREATION ALGORITHM

The efficient designs of a complementary set are depending on the generator (creator) and correlator complexity reduction. Therefore, to satisfy this requirement, there are some features must be identities, these are;

1. It has coupled complementary code sequences (A and B) with same length.
2. The output when use this code will get from eq's (3) and (4).
3. This output does not produce from unique A and B, But maybe get from code sequences (A) with more than one code sequence (B) and vice versa as indicated in Table (1).
4. Therefore, the selection of A and B will be depended on the applications.
5. Not all formations of selected bits are suitable as Golay code sequences.
6. Other formations which can be suitable as Golay code sequences, which satisfy the eq's (3) and (4) have the output as;
   a. Main loop =2N   Side-loop = 0 as in figure(2-a section 1 and 2-b section 5).
   b. Main loop = 2N   Side-loop ≠ 0 as in figure(2-a,b other sections).
7. The autocorrelation output of complementary codes A and B which gives sidelobe=0 can get from:

   $A \;\; B, -A \;\; B, A^{*} \;\; B, -A^{*} \;\; B, A \;\; B^{*}, -A \;\; B^{*}, A^{*} \;\; B^{*}, -A^{*} \;\; B^{*},$
   $A \;\; -B, -A \;\; -B, A^{*} \;\; -B, -A^{*} \;\; -B, A \;\; -B^{*}, -A \;\; -B^{*}, A^{*} \;\; -B^{*}, -A^{*} \;\; -B^{*}.$

   Where:

   -A =inverse of A.
   A^{*}= invert of A.
   $-A^{*}$= invert (reverse (A)) = reverse (invert (A)).
   Also, its same for B.

The proposed algorithm are based on 8-bit as shown;

Let A and B a complementary Golay code sequences, and

$$A = [A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1\ A_0];$$
$$-A = [-A_7\ -A_6\ -A_5\ -A_4\ -A_3\ -A_2\ -A_1\ -A_0];$$
$$A^{*} = [A_0\ A_1\ A_2\ A_3\ A_4\ A_5\ A_6\ A_7];$$
$$-A^{*} = [-A_0\ -A_1\ -A_2\ -A_3\ -A_4\ -A_5\ -A_6\ -A_7];$$

Also, for B are the same.

If $X = A \text{ XOR } (-A^{*}) \Rightarrow$ symmetry, and
$Y = B \text{ XOR } (-B^{*}) \Rightarrow$ symmetry.

Then; for complementary A and B we will obtain;

Y=inverse (X)
$Y = -X = B \text{ XOR } (-B^{*})$
$\therefore \; Y_0 = B_0 \text{ XOR } (-B^{*}_0) = B_0 \text{ XOR } (-B_7)$
$-X_0 = -(A_0 \text{ XOR } (-A^{*}_0))$
$= (-(A_0 \text{ XOR } (-A_7))$; and
$Y_1 = B_1 \text{ XOR } (-B_6)$
$Y_2 = B_2 \text{ XOR } (-B_5)$
$Y_3 = B_3 \text{ XOR } (-B_4)$

Then, we will get B codes, which represent the complementary of A from A, and to more reliable apply eq's (3) and (4).

SIMULATION AND IMPLEMENTATION RESULTS

In Figure (1), flowchart and VHDL program to generate and correlate a Golay complementary code, which based on eight 8-bits implementation.Table (1) represents 192 Golay pair, which can be generating while only 48 can used because each four codes give the same response as explain in the features and as showed in the Table. We checked our algorithm to generate the Golay code for 4, 8, 10, and 16-bits, the result as in Table (2). Then, we simulate the correlation of 8-bit code after generate the code by proposed algorithm by matlab 2012 for check its work according to the equations (3) and (4), and the result as in Figure (2 and 3), after that, implement the algorithm by Xilinx-Spartan-3A XC3S700A FPGA, and the result as in Figure (4) which represents the A code with its related Bs. Figure(5) represents the autocorrelation output which related to simulation as in Figure(3-b). The hardware implementation as in Figure (6) by Xilinx-Spartan-3A XC3S700A FPGA.

The auto-correlation checking was made to the sidelobe, because it's equal to zero for the complementary pair code only, and for other, pair will be not equal zero. While for the all, the main will be equal to 2N as in eq. 3 and in Figure (2) and (3). In Figure(2-a) the complementary code takes place in section 1, while in Figure(2-b) it takes place in section 5. Figure(3-a) represents the whole autocorrelation adding process for (A's) with all (B's). While Figure(3-b) represents the output of adding process when sidelobe checking has been satisfied, where, only complementary code exists. Figure(5), represents the output of the same operation with Xilinx-Spartan-3A XC3S700A FPGA implementation as in Figure(6). It's clear from the figure the output only for a complementary pair according to sidelobe checking as in Figure(6) with green indicators. The correct output will be obtained when the complementary pairs are conformal.

CONCLUSIONS

The proposed implementation designs are based on 8-bits after simulate 4,8,10 and 16 bit as shown in the results. This algorithm takes less time than checking all possible numbers to find the required complementary code by 1/4 for 4-bit, 1/16 for 8-bit, 1/248 for 10 bit and 1/1024 for 16 bits. The implementation could modify for any number of bits as required the application need. This flexibility in the implementation was obtained from the use of Xilinx –Spartan-3 XC3S700 FPGA.

REFERENCES

[1].Pace and C.Y. Ng, P.E. "Costas CW frequency hopping radar waveform: peak sidelobe improvement using Golay complementary sequences", Electronic Letts. 21st January 2010 Vol. 46 No. 2.

[2]. Zoltowski, M. D. T.R. Qureshi, R. Calderbank, and W. Moran, "Unitary Design of Radar Waveform Diversity Sets", in conf. Rec. Forty Second Asilomar Conf. Signal, Syst., Comput., Pacific Grove, CA, 2008.

[3].Jimenez, V.P.G., Fernandez, M.S.and Armada, A.G., " Study and implementation of complementary Golay sequences for PAR reduction in OFDM signals", IEEE, Electrotechnical Conf., Melecon, Pag. 198-203, 2002.

[4].Borislav Y. Bedahev, Zhaneta N. ZT. And Borislav P. S., " An algorithm for Computer Synthesis of Complementary Sequences", International Conference on System and Tchnologies-CompSys Tech, 2004.

[5].Carlos D. M. Jesus U. and Alvaro H., "Modular Architecture for Efficient Generation and Correlation of Complementary Set of Sequences", IEEE, Trans. On Signal Processing Vol. 55, No. 5, May 2007.

[6].Changho S., and Chan S. H., " Recursive Constraction of Golay Sequences", ISIT, USA, July 2007.

[7].Jonathan J. Matthew G. P., "A Construction of Binary Golay Sequences Pairs From Odd-Length Barker Sequences", Journal of Cmbinatorial Design, Vol. 17, Issue 6, Pag. 478-491, Nov. 2009.

```
a1:=A;
ad:= not a1;
for i in 0 to n1 loop
  x:=ad(i);ad(i):=ad(n-i);ad(n-i):=x;
end loop;
aa:=a1 xor ad;
bb:= not aa;
ind:=0;
for i in 0 to 15 loop
  x1(3 downto 0):= CONV_STD_LOGIC_VECTOR(i,4);
  for j in n1+1 to n loop
    x1(j):= bb(j) xor (not x1(n-j));
  end loop;
  xc:=conv(x1,n);
  ac:=conv(a1,n);
  cor(ac,xc,n,pp);
  zc:=0;
  for j in 0 to n loop
    if pp(j)=0 then
      zc:=zc+1;
    end if;
  end loop;
  if zc=n then
    ind:=ind+1;
    bx(ind):=x1;
  end if;
end loop;
B1<= bx(1);
B2<= bx(2);
B3<= bx(3);
B4<= bx(4);
```

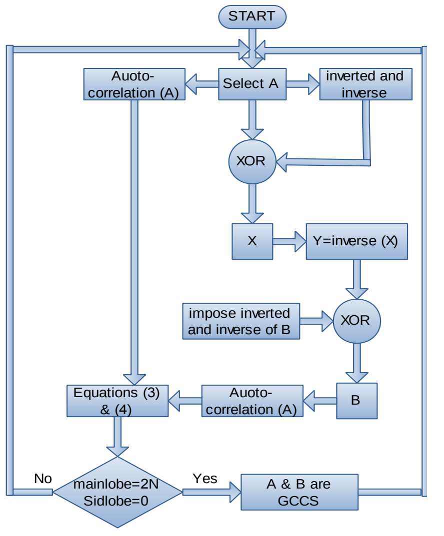
Reverse process

XOR & NOT process

16 iterations for determine Bs

```
procedure cor (x1,x2:in z;n:in integer; x3:out z) is
variable pa,pb:z;
for j in 0 to n loop
  pa(j):=0;
  pb(j):=0;
  for k in 0 to n-j loop
    pa(j):=pa(j)+x1(k)*x1(k+j);
    pb(j):=pb(j)+x2(k)*x2(k+j);
  end loop;
  x3(j):=pa(j)+pb(j);
end loop;
```

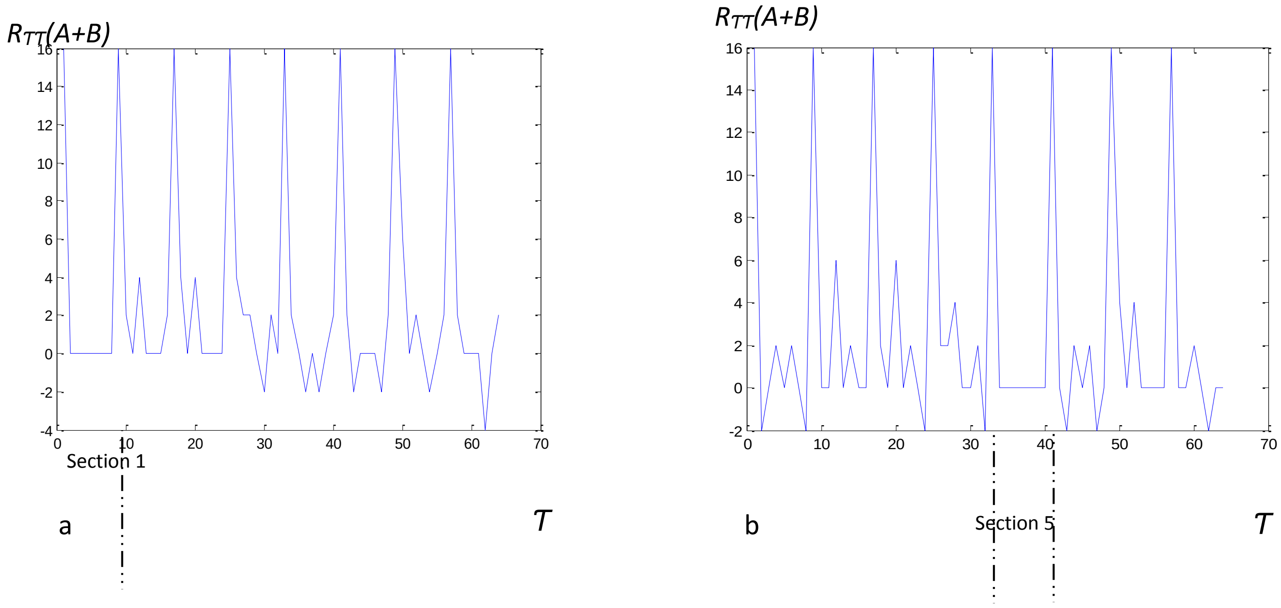
Autocorrelation process



a) flowchart of proposed algorithm

b) VHDL program for generate and correlate GCCS

Figure (1) Proposed algorithm for generate and correlate GCCS.



Figure(2) Output of equations (3,4) for complementary and not compatible complementary.

Table (1) Golay Code Sequences formation (8 bit).

A(Dec.)	B(Dec.)	A(Dec.)	B(Dec.)	A(Dec.)	B(Dec.)	A(Dec.)	B(Dec.)	A(Dec.)	B(Dec.)
172	6	144	58	177	125	141	190	202	249
83	6	111	58	114	125	78	190	53	249
53	6	141	65	114	130	9	197		
202	6	177	65	78	130	246	197		
197	9	114	65	141	130	111	197		
58	9	78	65	177	130	144	197		
163	9	18	71	116	132	96	202		
92	9	72	71	46	132	159	202		
184	18	237	71	139	132	249	202		
71	18	183	71	209	132	6	202		
29	18	226	72	132	139	222	209		
226	18	71	72	222	139	33	209		
27	20	184	72	123	139	123	209		
216	20	29	72	33	139	132	209		
39	20	125	78	130	141	39	215		
228	20	190	78	125	141	27	215		
20	27	65	78	65	141	228	215		
215	27	130	78	190	141	216	215		

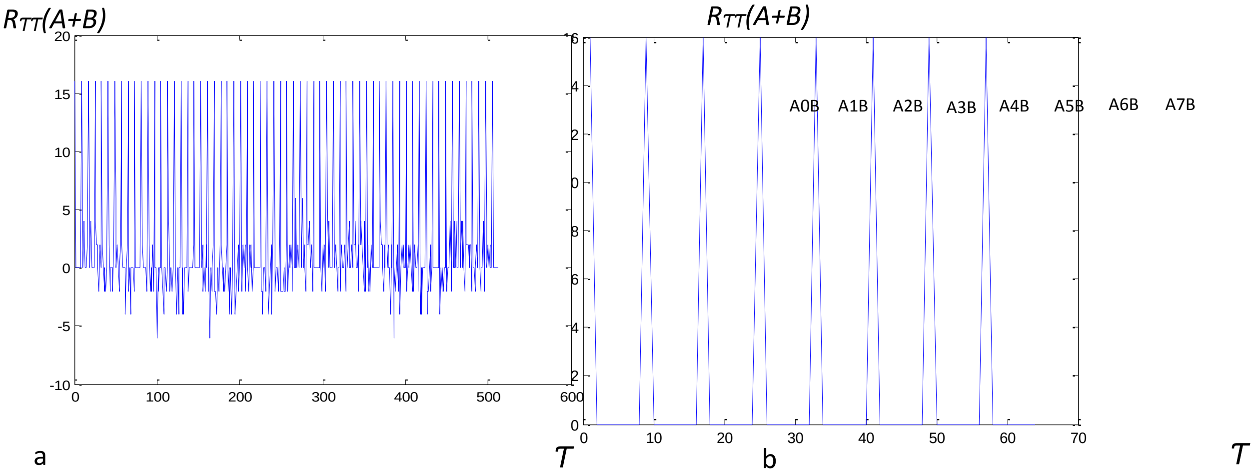
40	27	249	83	163	144	235	216		
235	27	6	83	197	144	20	216		
72	29	96	83	92	144	215	216		
18	29	159	83	58	144	40	216		
183	29	9	92	172	159	209	222		
237	29	111	92	83	159	46	222		
46	33	246	92	53	159	116	222		
209	33	144	92	202	159	139	222		
139	33	83	96	111	163	18	226		
116	33	172	96	9	163	72	226		
215	39	202	96	144	163	237	226		
235	39	53	96	246	163	183	226		
20	39	197	111	6	172	40	228		
40	39	163	111	249	172	215	228		
27	40	58	111	159	172	235	228		
39	40	92	111	96	172	20	228		
216	40	65	114	190	177	216	235		
228	40	125	114	65	177	27	235		
132	46	190	114	125	177	228	235		
222	46	130	114	130	177	39	235		
123	46	222	116	184	183	29	237		
33	46	33	116	226	183	71	237		
159	53	123	116	71	183	226	237		
96	53	132	116	29	183	184	237		
6	53	46	123	72	184	58	246		
249	53	116	123	18	184	197	246		
246	58	209	123	183	184	92	246		
9	58	139	123	237	184	163	246		
		141	125	177	190	83	249		
		78	125	114	190	172	249		

Table (2) Golay complementary for different number of bits

No. of bits	Total codes	Available codes
4	32	8
8	192	48
10	128	32
16	1536	384

Table(3) 8 bit Goly codes sequences (not repeated in A nor in B).

A(Dec.)	B(Dec.)
6	53
9	58
18	29
20	27
27	20
29	18
33	46
46	33
53	6
58	9
65	78
78	65



Figure(3). Output of equations (3,4): a) all output for each correlation process b) only sidelobe=0 output.

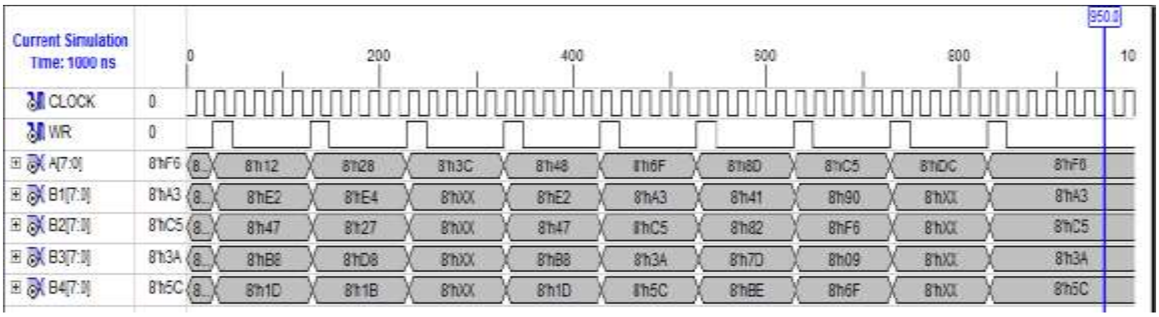
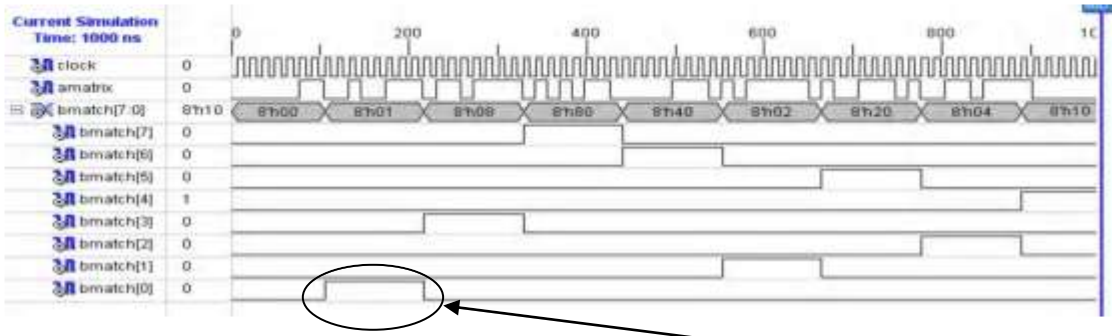
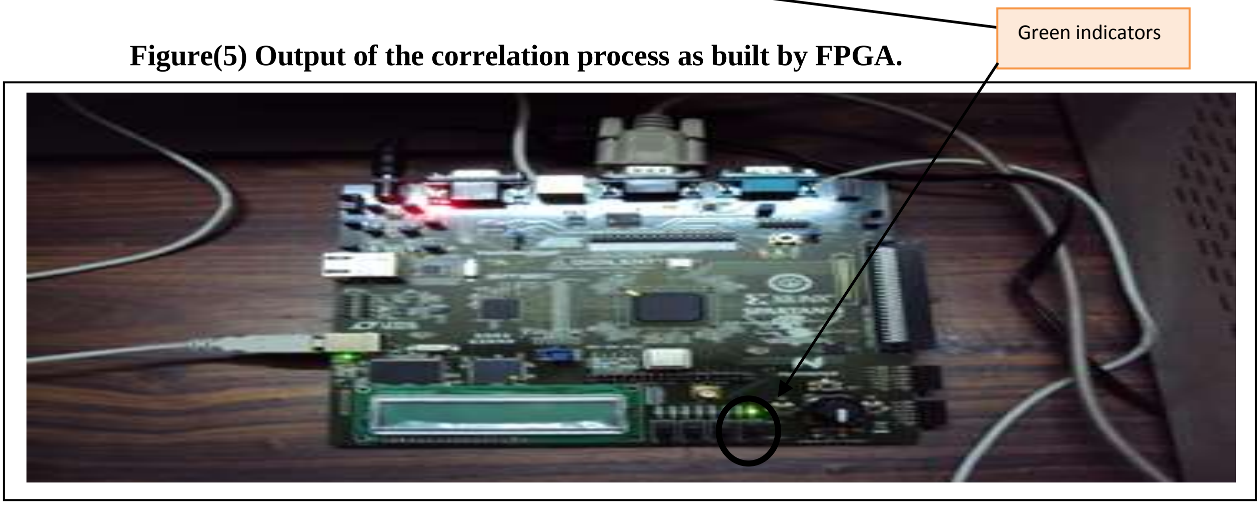


Figure (4) generate complementary B for A and from A.



Figure(5) Output of the correlation process as built by FPGA.



Figure(6) Implementation Circuit for Generate and Correlate of GCCS.