4-Bit Pseudo Random Sequence Generator and Detector in FPGA

Mrinal Senapati, Vivek Barsaiyan †

***Abstract*—We implemented a 4-bit Pseudo Random Sequence Generator and Detector circuit in FPGA board.For this purpose, We are using IcoBoard which is FPGA board with at max 100MHz clock frequency with up to 8 MBit of SRAM and is programmable in Verilog [3].In IcoBoard, we are generating the 4-bit Random Sequence using LFSR (Linear Feedback Shift Register) continuously with 3 seconds delay. User can give any 4-bit sequence using the keypad and depending upon detection is successful or not,output will be shown through leds and also in LCD display.**

I. Objective

Implementing 4-bit Pseudo Random Sequence Generator and Detector circuit in FPGA.

II. Equipment List

A. Hardware Requirements

1) FPGA based i/o icoBoard (x1)
2) RaspberryPi-Jessie (x1)
3) Arduino UNO (x2)
4) LCD 16x2 i2c display (x1)
5) Keypad 4x3 (x1)
6) Potentio Metre (x1)
7) Breadboard (x2)
8) LEDS (3 different color (1,4,4 respectively))
9) Jumper wires

B. Software Requirements

1) Raspberry pi equipped with wiringPi, icoProg and icoStorm.
2) Arduino version 1.8x
3) Verilog

III. Theory

The project is divided with 3 stages :

1) Entering 4 bit input sequence to the IcoBoard using Keypad and Arduino UNO.
2) Verilog programming for generating 4-bit pseudo random bit sequence and detecting it.
3) Showing the generated 4-bit sequence and detected output using LCD Display and Arduino UNO.

We are using **4x3 Keypad** for giving the input bit sequence to be detected.User is asked to enter any nunber from 0 to 15 which then will be converted into 4-bit binary sequence in the Ardino.This 4-bit then will be given to the Ico-Board.

A **pseudorandom number generator (PRNG)** also known as a deterministic random bit generator (DRBG) [1] is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The PRNG-generated sequence is not truly random, because it is completely determined by an initial value, called the PRNG's seed (which may include truly random values). Although sequences that are closer to truly random can be generated using hardware random number generators, pseudorandom number generators are important in practice for their speed in number generation and their reproducibility [2].

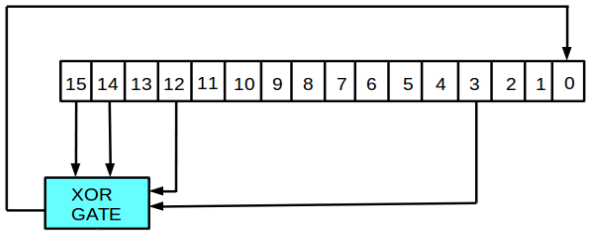

Figure 1: **Linear Feedback Shift Register** for generating Pseudo Random Bit

Linear Feedback Shift Register(LFSR) has been used for generating the pseudo random bits.The most commonly used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value.Here we took decimal value **15** for the LFSR'seed and xored

†The authors are with the Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Hyderabad 502285, India (e-mail:**ee18mtech11025@iith.ac.in; ee18mtech11028@iith.ac.in**)

3,12,14,15 bits to generate the random bits and stored in the LSB(see fig.1).

Bit-wise comparison technique is used for detecting input sequence with the generated pseudo random bit sequence.

After detecting the sequence, output is shown in **LCD display** with the generated pseudo random bit sequence. Output will be show as '**1**' for successful detecting the sequence else '**0**'.

IV. Procedure

A. Keypad-Arduino Interface:

1) Build the Arduino Keypad connection using the Hardware components and circuit diagram referred in fig 2.
   Note-Digital pins of arduino connected to icoborad.
2) Write the arduino code referred in code table 1.

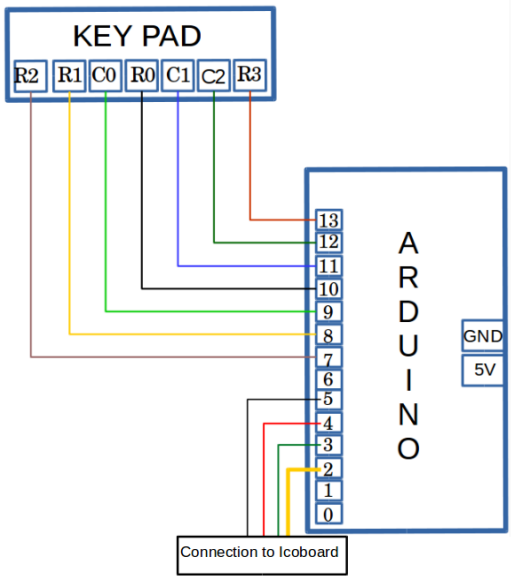

Figure 2: Arduino UNO-Keypad(4x3) Connections

B. LCD display-Arduino Interface:

1) Build the Arduino Keypad connection using the Hardware components and circuit diagram referred in fig 3.
   Note-Digital pins of arduino connected to icoborad.
2) Write the arduino code referred in code table 2.

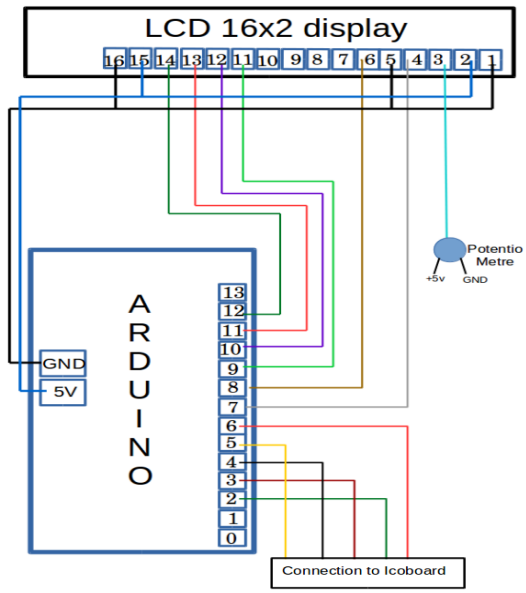

Figure 3: Arduino UNO-LCD display(16x2) Connections

C. Icoboard-Arduino Interface:

1) Connect the above two setups (A,B) of arduino with the icoboard using digital pins of arduino referred in fig 4.
2) Write the verilog code in FPGA icoboard referred in code table 3.

V. Arduino and Verilog Programming:

A. Arduino Code for Giving input from keypad:

Here user can give input from the keypad in between 0-15 any number.Entered number is then converted into 4-bit sequence and sent to icoboard.If the user give any number more than 15 ,it will ask again to enter a number in between 0-15.

```
#include <Keypad.h>
float v1 = 0;
float v2 = 0;
int a, b, c, d;
int bits[4];
const byte ROWS = 4; //four rows
const byte COLS = 3; //three columns
char keys[ROWS][COLS] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'.', '0', '#'}
};
```

```
byte rowPins[ROWS] = {10, 8 , 7, 13};
byte colPins[COLS] = { 9, 11 , 12};
Keypad kpd = Keypad( makeKeymap(keys),
rowPins, colPins, ROWS, COLS );
void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  Serial.begin(9600);
}
byte value = 0;
void loop() {
char key = kpd.getKey();
if (key != NO_KEY)
{
if ( (key >= '0') && (key <= '9') )
{
value = value * 10;
value = value + key - '0';
}
if ( key == '#' )
{
delay(1000);
if (value <= 15) {
Serial.print("The data is : ");
Serial.println(value);
int data = value;
for (int c = 3; c >= 0; c--)
{
int k = data >> c;
if (k & 1) {
bits[c] = 1;
}
else {
bits[c] = 0;
}
}
Serial.print("Bits sent to fpga board:
");
Serial.print(bits[3]);
Serial.print(bits[2]);
Serial.print(bits[1]);
Serial.println(bits[0]);
}
else {
Serial.println("Plaese enter a value
less
than 16!!");
}
value = 0; //Now reset ready for next
input
}
}
  digitalWrite(5, bits[3]);
  digitalWrite(4, bits[2]);
  digitalWrite(3, bits[1]);
  digitalWrite(2, bits[0]);

}
```

B. Arduino code for LCD display:

Here output coming from the icoboard are printed in the LCD display.Arduino is taking 5 bit input from the icoboard (4-bit for pseudo random sequence and 1 bit for show output) and printing the output in LCD screen.

```
#include<LiquidCrystal.h>
// initialize the library by
associating any needed LCD interface
pin// with the arduino pin number
it is connected to
const int rs = 7, en = 8,
d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4,
d5, d6, d7);
String a;
int temp=0;
void setup() {
pinMode(2, INPUT);  // connect to R10
pinMode(3, INPUT);
pinMode(4, INPUT);
pinMode(5, INPUT);   //connect to T15
pinMode(6,INPUT);
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Random Sequence is ");
Serial.begin(9600);
// opens serial port, sets data
rate to 9600 bps
}
void loop() {
temp=0;
int a;
Serial.println(a);
lcd.setCursor(0,1);
lcd.print(a);
int b=digitalRead(3);
lcd.setCursor(1,1);
lcd.print(b);
int c=digitalRead(4);
lcd.setCursor(2,1);
lcd.print(c);
```

```
int d=digitalRead(5);
lcd.setCursor(3,1);
lcd.print(d);
//Serial.println(String(temp));
int e=digitalRead(6);
lcd.setCursor(14,1);
lcd.print(e);
//delay(2000);

}
```

C. Verilog Code :

In this code,LFSR(Linear Feedback Shift Register) is used for pseudo random bit generation.We are generating 4 bits sequence using LFSR and then comparing the generated bits with the input given by the user.

```
module seven_segment(clk,e,f,g,h,a,b,c,
d,i,j,k,l,out_led,i1,j1,k1,l1,out1);
input wire clk,e,f,g,h;
output reg a,b,c,d,i,j,k,l,out_led;
output reg i1,j1,k1,l1,out1;
reg [28:0] delay;
reg [3:0] bits;
reg [15:0] rand_num;
wire feedback;
initial begin
a=1;
b=1;
c=1;
d=1;
bits=4'b0000;
rand_num=16'd15;
out_led=0;
end
always@(posedge clk)
begin
delay=delay+1;
if(delay==29'b1000111100
0011010001100000000) //3 seconds delay
begin
delay=29'b0;
//sending data from arduino
a=e;
b=f;
c=g;
d=h;
//generating random bits
feedback = rand_num[15] ^ rand_num[14]
^ rand_num[12] ^ rand_num[3];
rand_num={rand_num[14:0], feedback};
bits[0]=rand_num[0];
feedback = rand_num[15] ^ rand_num[14]
^ rand_num[12] ^ rand_num[3];
rand_num={rand_num[14:0], feedback};
bits[1]=rand_num[0];
feedback = rand_num[15] ^ rand_num[14]
^ rand_num[12] ^ rand_num[3];
rand_num={rand_num[14:0], feedback};
bits[2]=rand_num[0];
feedback = rand_num[15] ^ rand_num[14]
^ rand_num[12] ^ rand_num[3];
rand_num={rand_num[14:0], feedback};
bits[3]=rand_num[0];
i=bits[0];
j=bits[1];
k=bits[2];
l=bits[3]
//Going into lcd display
i1=bits[0];
j1=bits[1];
k1=bits[2];
l1=bits[3];
//comparing bits for detecting sequence
if(a==i && b==j && c==k && d==1)
begin
out_led=1;
out1=1;
end
else
begin
out_led=0;
out1=0;
end
end
end
endmodule
```

VI. Conclusion

1) Working principle of LFSR circuit is implemented in FPGA board.
2) User friendly interface is implemented for giving input from keypad directly.
3) Output is shown using LCD display and also with the help colorful leds.

References

[1] Barker E., Kelsey J., Recommendation for Random Number Generation Using Deterministic Random Bit Generators, NIST SP800-90A, January 2012

[2] "Pseudorandom number generators". Khan Academy. Retrieved 2016-01-11.

[3] http://icoboard.org/

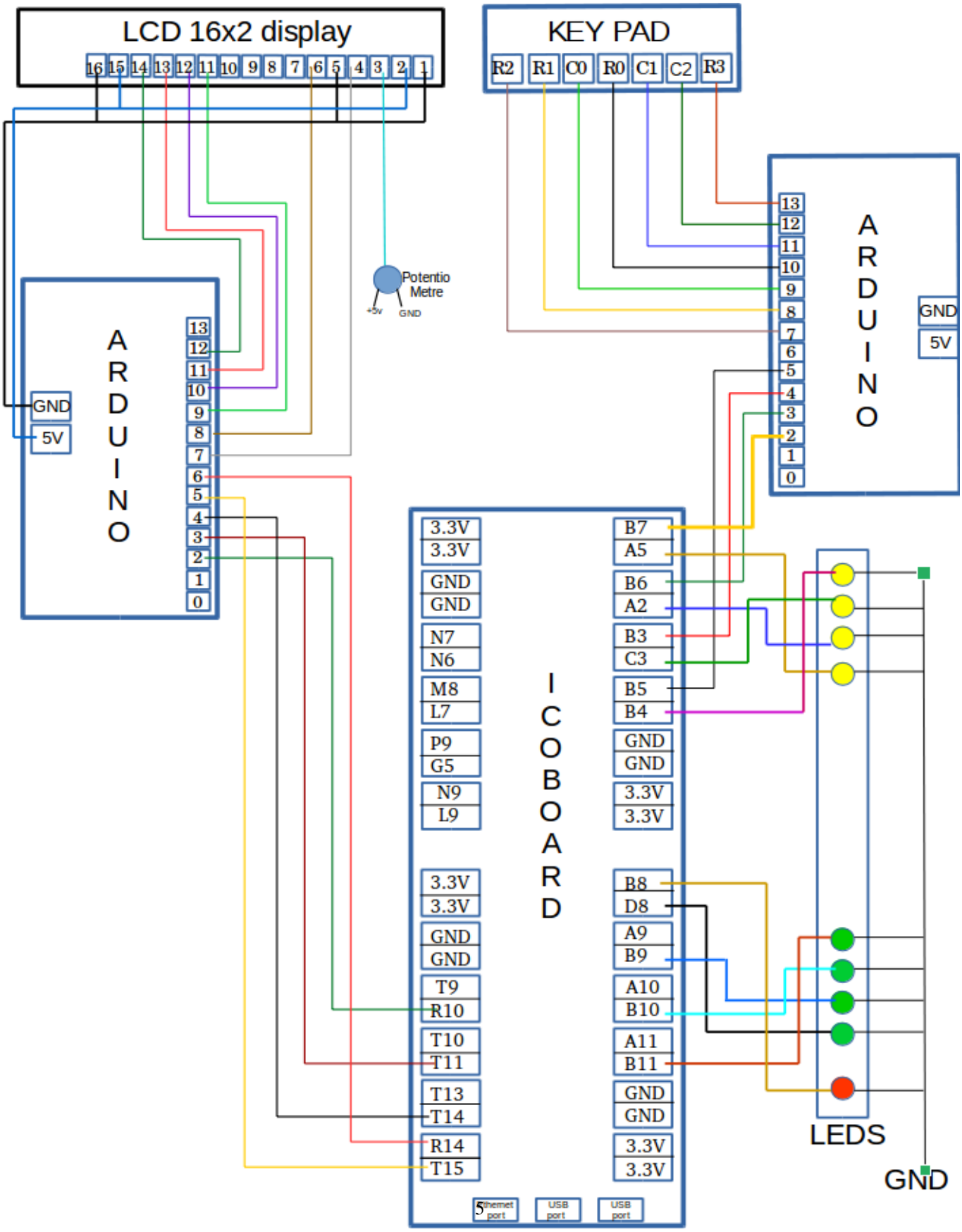

Figure 4: Hardware Connections