



ITMO UNIVERSITY

**Университет ИТМО.
Отчет по лабораторной работе номер 4
по дисциплине «Функциональная схемотехника»**

Выполнил : Дробыш Дмитрий Александрович
Группа : Р33082
Преподаватель : Кустарев Павел Валерьевич

Цель работы.

Получить навыки разработки встроенных схем тестирования (BIST) для цифровых устройств, реализованных на базе программируемых логических интегральных схем.

Задание.

Вариант	Формула	Lfsr1	Lfsr2	8 CRC
5	$3a + 2 \cdot b^{\frac{1}{3}}$	$1 + x + x^6 + x^7 + x^8$	$1 + x^4 + x^6 + x^7 + x^8$	$1 + x^4 + x^5 + x^6 + x^8$

Таблица 1. Вариант на лабораторную работу 4.

Схемы модулей.

LFSR1:

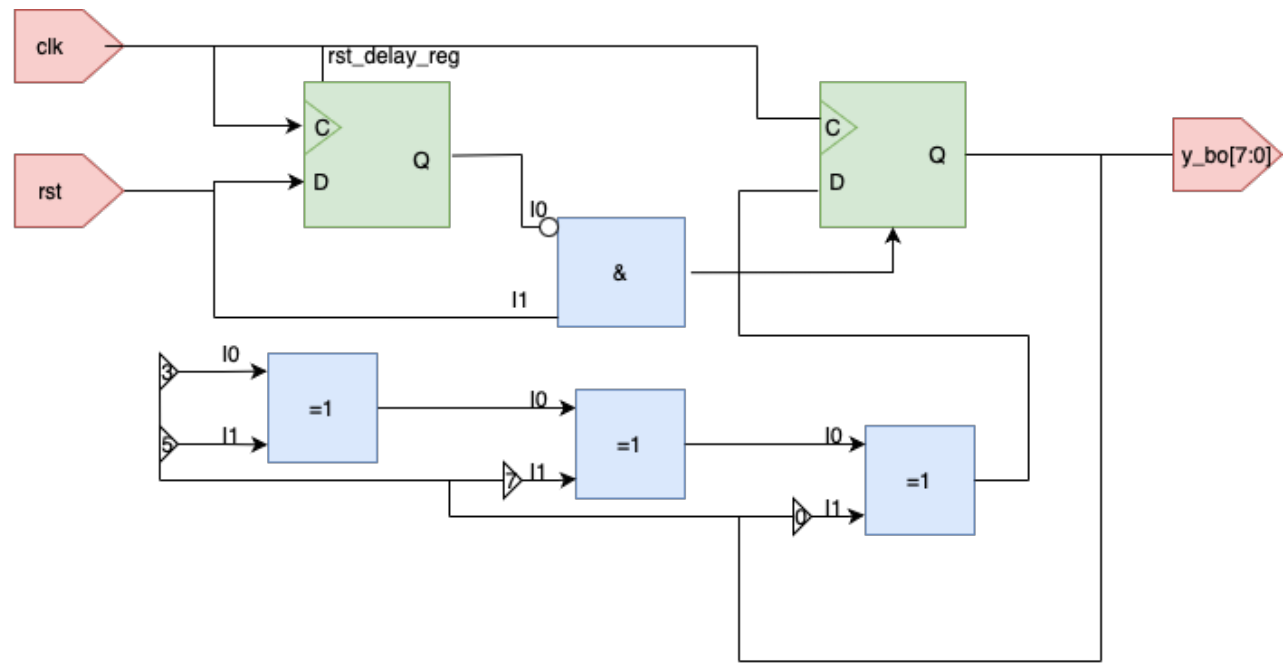


Рис. 1. Модуль LFSR.

LFSR2 :

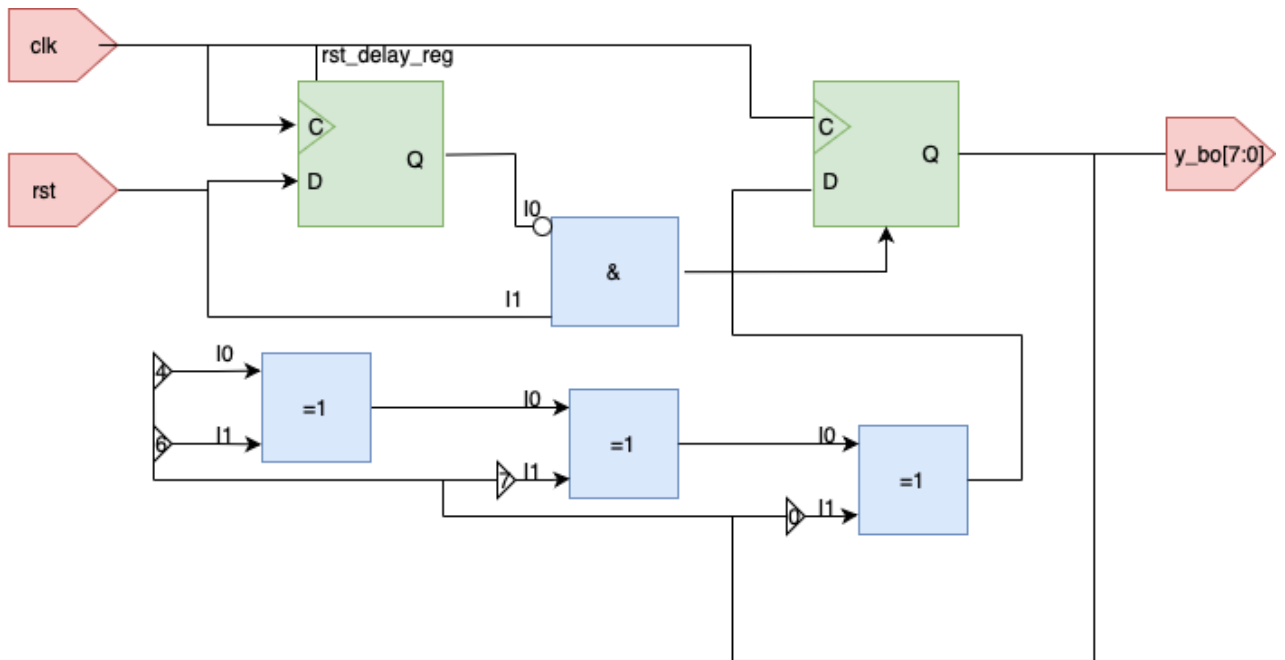


Рис.2. Модуль LFSR2.

BIST:

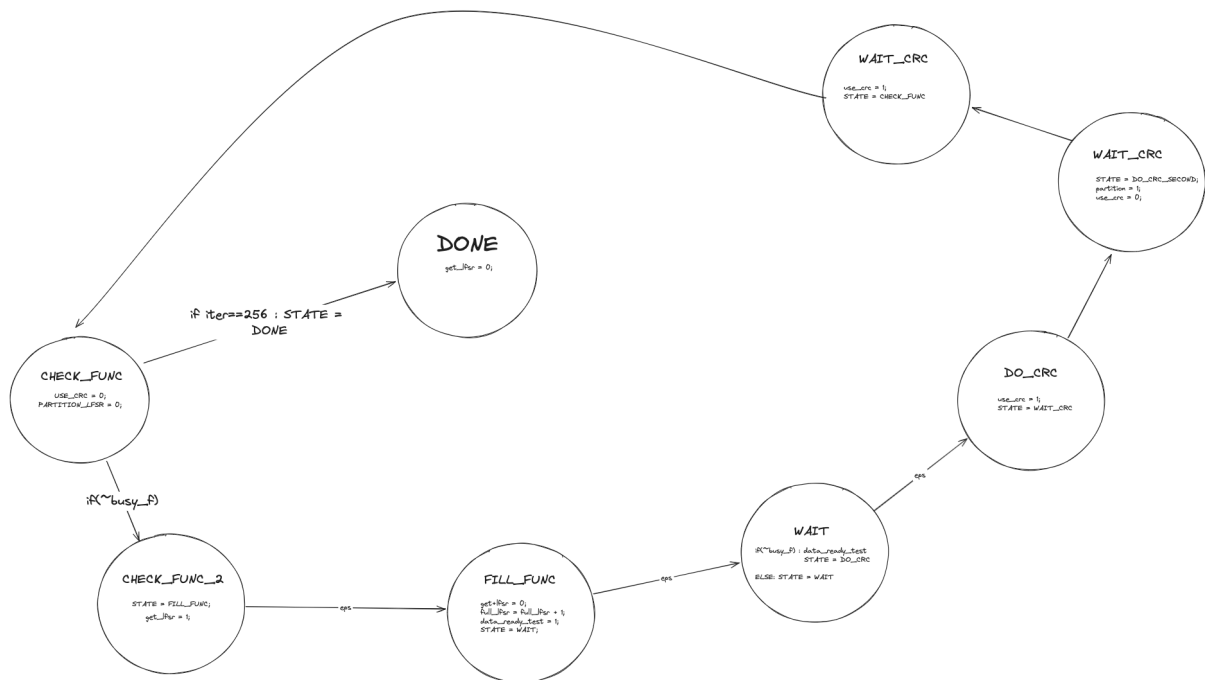


Рис. 3. Модуль BIST

Плотность распределения значений LFSR

График с плотностью распределения значений операндов, которые перебираются в режиме самотестирования, с указанием области допустимых значений:

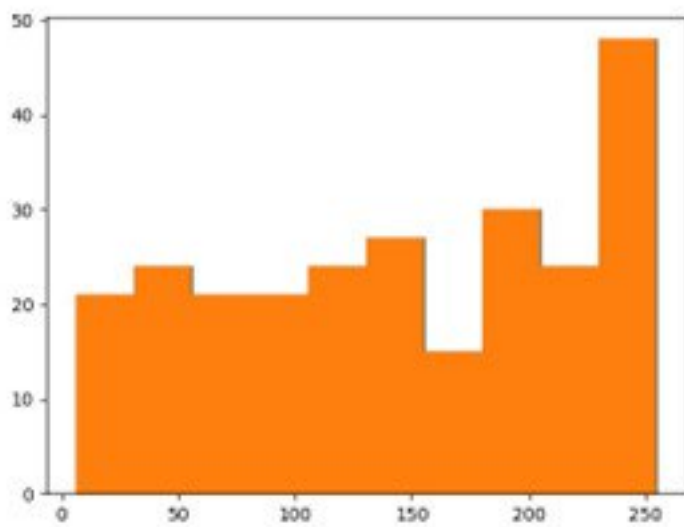


График 1.LFSR2

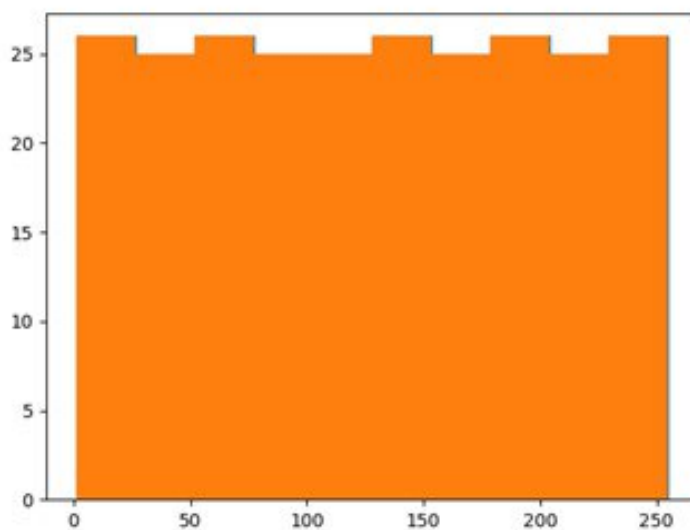


График 2. LFSR1

ОДЗ : $X \in [0; 255]$ (8 bit)

Процент значений, попавших в ОДЗ — 100%

Результат тестирования блока в icarus Verilog test bench:

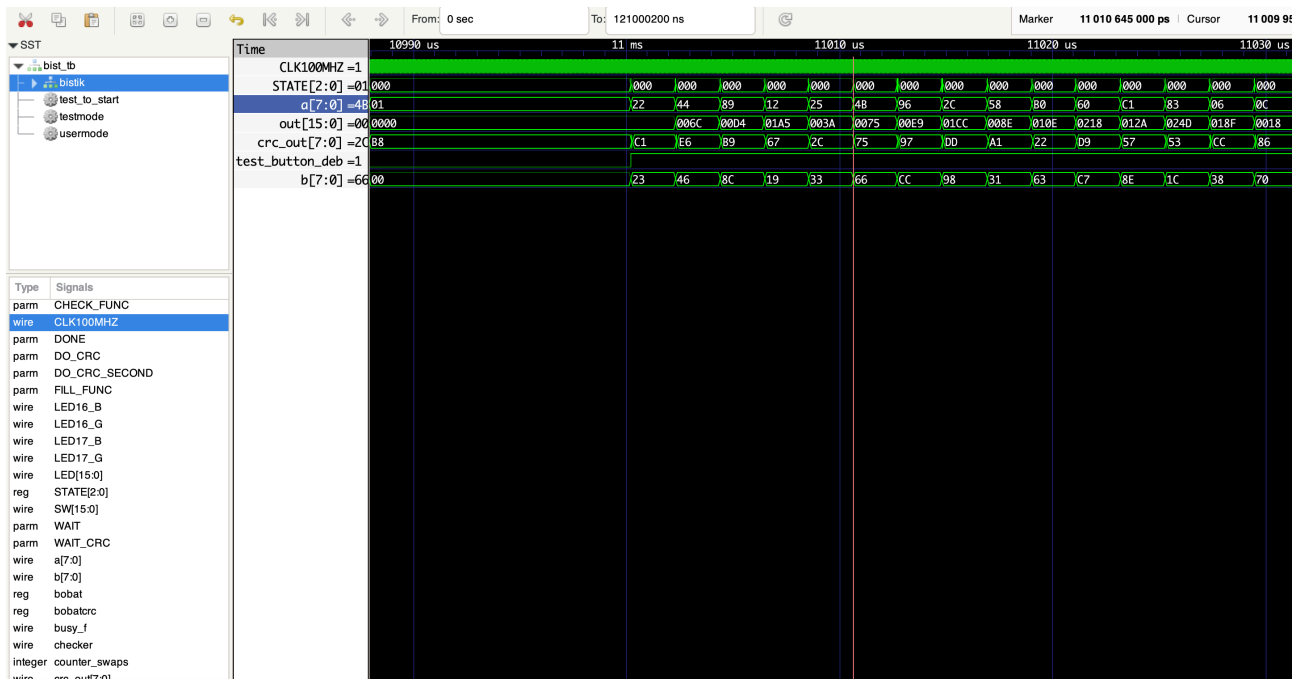


Рис. 4. Тестирование модуля bist.

Подробно:

VCD info: dumpfile bist.vcd opened for output.

usermode check

Pressed RST : 0

Pressed RST : 0

Pressed START : 0

Answer is 3

test check

Pressed RST : 0

Unpressed RST : 0

Pressed TEST : 89

Unpressed TEST : 89

Pressed TEST : 0

Unpressed TEST : 0

Pressed TEST : 89

Unpressed TEST : 89

Pressed TEST : 0

Unpressed TEST : 0

Pressed TEST : 89

Unpressed TEST : 89

Answer is 89

test to start check

Pressed RST : 89

Unpressed RST : 89

Pressed TEST : 0

Pressed START : 0

Unpressed START : 0

Unpressed TEST : 0

Pressed TEST : 89
Unpressed TEST : 89
Pressed START : 89
Unpressed START : 89
Pressed TEST : 0
Unpressed TEST : 0
Answer is 0
bist_tb.v:27: \$finish called at 121000200000 (1ps)

Время работы алгоритма составляет : 526.62 mcs.

Верификация работы алгоритма.

Для верификации работы модуля был написан соответствующий тестовый код на языке программирования C. Программа сравнивает результаты работы схемы с теоретическими данными.

Code snippet:

```
#include <stdint.h>
#include <stdio.h>
#include <math.h>
```

```
uint8_t crc_my(uint8_t sum, uint8_t in){
    uint8_t shift = sum;
    for(uint8_t i = 0; i < 8; i++){
        shift ^= in;
        if(shift & 0b10000000){
            shift = (shift << 1);
            shift ^= 0b01110001;
        }else{
            shift = (shift << 1);
        }
    }
    return shift;
}
```

```
int main(void){
```

```
    uint8_t lfsra[255];
    uint8_t lfsrb[255];

    printf("{");
    for(uint64_t i = 0; i < 255; i++){
        printf("%#02x, ", 3*lfsra[i] + 2 *
((uint8_t)cbrt((double)lfsrb[i])));
    }
}
```

```

printf("}\n");

uint16_t testarr[256];
uint8_t crcval = 0xB8;
for(uint64_t i = 0; i < 256; i++){
    crcval = crc_my(crcval, (uint8_t)testarr[i]);
    crcval = crc_my(crcval, *((uint8_t*)(testarr+i
+1) ));
    printf("%d\n", crcval);
}

uint8_t arr_verilog[256];

for(uint64_t i = 0; i < 255; i++){
    if(arr_test[i] == arr_verilog[i]){
        printf("%ld - OK, v:%#02x, t:%#02x\n", i,
arr_verilog[i], arr_test[i]);
    }else{
        printf("!!! %ld - FAIL, v:%#02x, t:%#02x\n", i,
arr_verilog[i], arr_test[i]);
    }
}
}

```

Результаты работы:

```

0 - OK, v:0xb8, t:0xb8
1 - OK, v:0xc1, t:0xc1
2 - OK, v:0xe6, t:0xe6
3 - OK, v:0xb9, t:0xb9
4 - OK, v:0x67, t:0x67
5 - OK, v:0x2c, t:0x2c
6 - OK, v:0x75, t:0x75
7 - OK, v:0x97, t:0x97
8 - OK, v:0xdd, t:0xdd
9 - OK, v:0xa1, t:0xa1
10 - OK, v:0x22, t:0x22
11 - OK, v:0xd9, t:0xd9
12 - OK, v:0x57, t:0x57
13 - OK, v:0x53, t:0x53
14 - OK, v:0xcc, t:0xcc
15 - OK, v:0x86, t:0x86
16 - OK, v:0x8e, t:0x8e
17 - OK, v:0xe7, t:0xe7
18 - OK, v:0x17, t:0x17
19 - OK, v:0xcf, t:0xcf
20 - OK, v:0x59, t:0x59

```

...

Ref Name	Used	Functional Category
FDRE	261	Flop & Latch
LUT5	143	LUT
LUT6	98	LUT
LUT3	64	LUT
LUT4	55	LUT
LUT2	24	LUT
IBUF	21	IO
CARRY4	20	CarryLogic
OBUF	19	IO
MUXF7	17	MuxFx
LUT1	6	LUT
FDSE	4	Flop & Latch
OBUFT	1	IO
MUXF8	1	MuxFx
BUFG	1	Clock

Таблица 2. Primitives

Site Type	Used	Fixed	Available	Util%
BUFGCTRL	1	0	32	3.13

Таблица 3. Clocking.

Site Type	Used	Fixed	Available	Util%
LUT as Logic	336	0	63400	0.53
Register as Flip Flop	265	0	126800	0.21
F7 Muxes	17	0	31700	0.05
F8 Muxes	1	0	15850	<0.01

Таблица 4. Slice Logic.

Вывод:

В ходе выполнения лабораторной работы была доработана схема арифметического блока, а также реализованы: регистр сдвига с линейной обратной связью, алгоритм циклического избыточного кода, а также built-in-self-test механизм. В процессе работы возникли трудности, в частности с разделением панели вывода между компонентами системы и проблема наложения реакций на нажатие разных кнопок устройства. Для повышения доверия к качеству блоков были реализованы тесты на языке программирования C.