



Escuela de Ingeniería en Computadores

CE 4301 — Arquitectura de Computadores I

Simuladores ISA y Debug

Tarea 1

Profesor: Luis Chavarria Zamora

Estudiantes:

Carlos Andrés Mata Calderón

I Semestre 2024

LFSR

Se implementa el algoritmo de LFSR, para 3 arquitecturas diferentes. Se usa como semilla el valor $0x43 = 0100\ 0011$ qué es la letra C en ASCII. Se calcula con el polinomio de $x^8 + x^6 + x^5 + x^4 + 1$, y realiza se $steps=10$, y se almacenan en la dirección de memoria $0x100$ estos 10 valores.

Se usa la página <https://www.dcode.fr/linear-feedback-shift-register> para obtener los resultados esperados.

The image shows a web application titled "LINEAR FEEDBACK SHIFT REGISTER" with a breadcrumb trail "Informatics > Linear Feedback Shift Register". The main section is "LFSR BITS GENERATOR". Under "LFSR IN FIBONACCI MODE", the "INITIAL STATE OF THE REGISTER (IN BINARY)" is set to "0100 0011". There are controls for "Automatic Detection" and bit toggles for 0 and 1. The "BITS TO XOR (START AT 0 FROM THE LEFT)" are set to "3,4,5,7". Three radio buttons are present: "GENERATE N BITS, NUMBER OF CYCLES/SHIFTS N=" (set to 4,4), "FIND THE PERIOD (MAX 2^16)", and "SHOW FIRST ITERATIONS IN DETAILS" (which is selected). A "CALCULATE" button is at the bottom. A footer link says "See also: Random Selection — XOR Cipher".

Y se obtiene que para la step 10 el valor es **0x63**

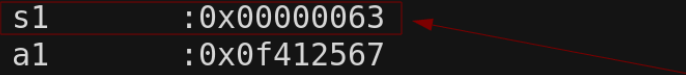
	A	B	C	D
1	step	bin	output	hex
2	0	1000011		43
3	1	10100001	1	A1
4	2	11010000	1	D0
5	3	11101000	0	E8
6	4	11110100	0	F4
7	5	1111010	0	7A
8	6	111101	0	3D
9	7	11110	1	1E
10	8	10001111	0	8F
11	9	11000111	1	C7
12	10	1100011	1	63
13	11	10110001	1	D1

Para comprobar que efectivamente el código funciona, se revisarán los registros para cada una de las arquitecturas. Además cada arquitectura tiene **un archivo de código** que tiene extensión **.s** para el caso de RISC-V y ARM, para x86-64 es **.asm**, **un archivo ejecutable** que tiene el ejecutable para esa arquitectura y **por último un archivo disassembly** que tiene extensión **.txt** que tiene las instrucciones desarmadas.

RISCV

El resultado para se almacena en s1. Se recomienda usar [Ripes Emulator](#) , para ver los 10 valores almacenados en memoria del algoritmo.

```
wc> rv-jit -d riscv
(rv-sim) break 0x1007c
.breakpoint 0x1007c
(rv-sim) run
(rv-sim) reg
instret : 3 time :0x00019ed9623b7cb7
pc      :0x0001007c fcsr :0x00000000
ra      :0x2bb0da50
sp      :0x7fffffff64 gp :0x0e4ea7f9
tp      :0x4ec07281 t0  :0x4fd278e4
t1      :0x46f35869 t2  :0xed33d65e
s0      :0xaf5b006e s1  :0x00000063
a0      :0x00000000 a1  :0x0f412567
a2      :0x3e5ec04f a3  :0x809d72b3
a4      :0x5c991f58 a5  :0x08d23516
a6      :0x8f6f53a6 a7  :0x00000000
s2      :0x1917867d s3  :0xc2003b71
s4      :0x8e2b8e36 s5  :0xab3ec81c
s6      :0x436a6013 s7  :0x1ad6742d
s8      :0xb4259a99 s9  :0x0d9a044d
s10     :0xb8a77a6c s11 :0x0e61cde6
t3      :0x5ccf39e9 t4  :0x394dba4e
t5      :0x707c7692 t6  :0x8ca93e90
(rv-sim) □
```



x86_64

El resultado para se almacena en r12 el valor.

```
> gdb x86
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.h
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from x86...
(No debugging symbols found in x86)
(gdb) b _end_for_memory_saver
Breakpoint 1 at 0x401030
(gdb) r
Starting program: /home/carlos/Repos/arch1_homework1/src/x86/x86

Breakpoint 1, 0x0000000000401030 in _end_for_memory_saver ()
(gdb) i r
rax            0x8            8
rbx            0x0            0
rcx            0x1            1
rdx            0x0            0
rsi            0x0            0
rdi            0x0            0
rbp            0x0            0x0
rsp            0x7fffffff270    0x7fffffff270
r8             0x0            0
r9             0x0            0
r10            0x0            0
r11            0x1            1
r12            0x63           99
r13            0x0            0
r14            0x0            0
r15            0x0            0
rip            0x401030        0x401030 <_end_for_memory_saver>
eflags         0x202          [ IF ]
cs             0x33           51
ss             0x2b           43
ds             0x0            0
es             0x0            0
fs             0x0            0
gs             0x0            0
(gdb) 
```

ARM

El resultado para se almacena en r5. Se recomienda usar [CPULator Simulator](#), para ver los 10 valores almacenados en memoria del algoritmo.

```
> cd arm
/(> qemu-arm -g 12345 ./arm &
[1] 100601
> gdb-multiarch arm
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from arm...
(No debugging symbols found in arm)
(gdb) target remote :12345
Undefined command: "target". Try "help".
(gdb) target remote :12345
Remote debugging using :12345
0x00010074 in _start ()
(gdb) b _end_for_memory_saver
Breakpoint 1 at 0x10130
(gdb) run
The "remote" target does not support "run". Try "help target" or "continue".
(gdb) c
Continuing.

> Breakpoint 1, 0x00010130 in _end_for_memory_saver ()
(gdb) i r
r0          0x20168      131432
r1          0x28         40
r2          0x0          0
r3          0x0          0
r4          0xff        255
r5          0x63         99
r6          0x20         32
r7          0x0          0
r8          0x10         16
r9          0x0          0
r10         0x20150     131408
r11         0x0          0
r12         0x0          0
sp          0x40800510   0x40800510
lr          0x0          0
pc          0x10130     0x10130 <_end_for_memory_saver>
```

