



# Versatile counter

---

**A reconfigurable binary, gray or LFSR counter**

**Brought to You By ORSoC / OpenCores**

## Legal Notices and Disclaimers

---

### Copyright Notice

This ebook is Copyright © 2009 ORSoC

### General Disclaimer

The Publisher has strived to be as accurate and complete as possible in the creation of this ebook, notwithstanding the fact that he does not warrant or represent at any time that the contents within are accurate due to the rapidly changing nature of information.

The Publisher will not be responsible for any losses or damages of any kind incurred by the reader whether directly or indirectly arising from the use of the information found in this ebook.

This ebook is not intended for use as a source of legal, business, accounting, financial, or medical advice. All readers are advised to seek services of competent professionals in the legal, business, accounting, finance, and medical fields.

No guarantees of any kind are made. Reader assumes responsibility for use of the information contained herein. The Publisher reserves the right to make changes without notice. The Publisher assumes no responsibility or liability whatsoever on the behalf of the reader of this report.

### Distribution Rights

The Publisher grants you the following rights for re-distribution of this ebook.

[YES] Can be given away.

[YES] Can be packaged.

[YES] Can be offered as a bonus.

[NO] Can be edited completely and your name put on it.

[YES] Can be used as web content.

[NO] Can be broken down into smaller articles.

[NO] Can be added to an e-course or auto-responder as content.

[NO] Can be submitted to article directories (even YOURS) IF at least half is rewritten!

[NO] Can be added to paid membership sites.

[NO] Can be added to an ebook/PDF as content.

[NO] Can be offered through auction sites.

[NO] Can sell Resale Rights.

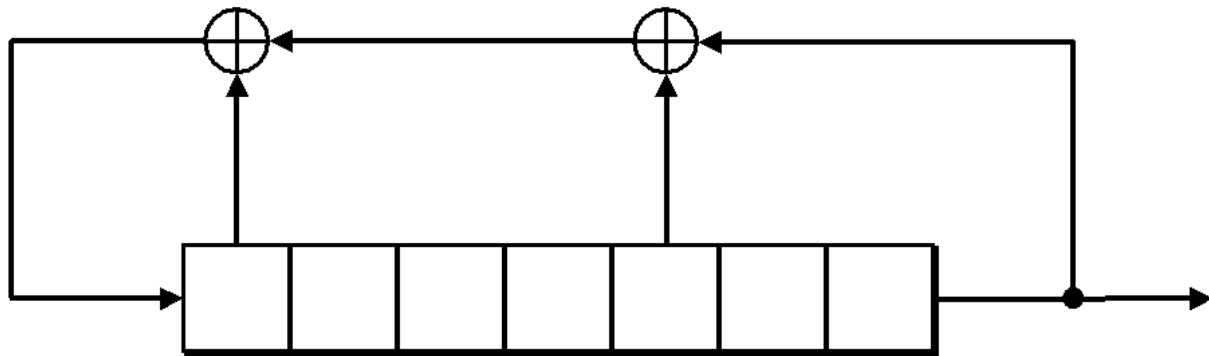
[NO] Can sell Master Resale Rights.

[NO] Can sell Private Label Rights.

## Table of Contents

<b>Chapter 1 Linear Feedback Shift Registers</b>	<b>4</b>
<b>Chapter 2 Implementation</b>	<b>5</b>
Usage notes	8
<b>Appendix A:</b>	
<b>excel2csv</b>	<b>9</b>
<b>Recommended Resources</b>	<b>10</b>

## Chapter 1 Linear Feedback Shift Registers



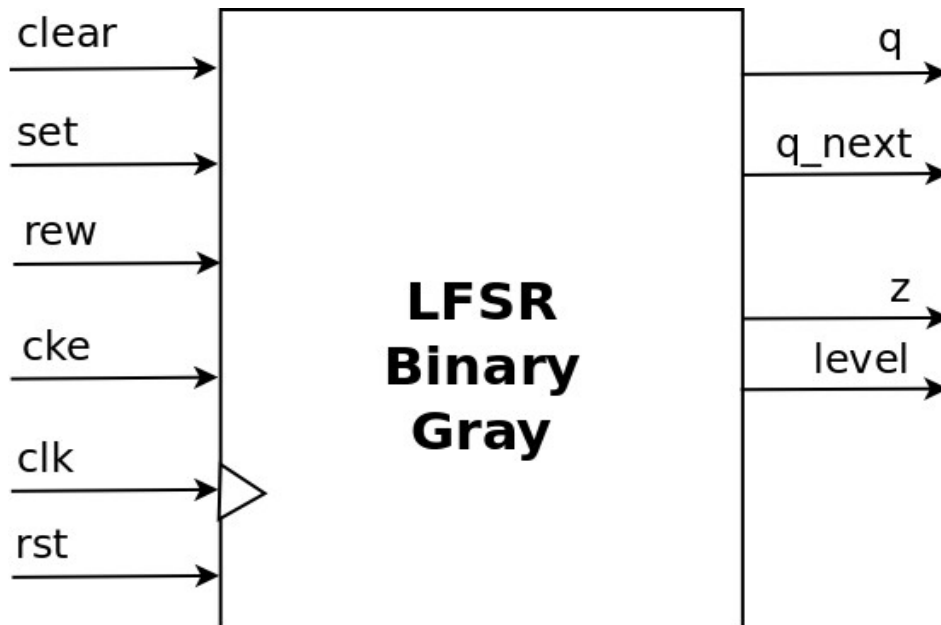
A **linear feedback shift register** (LFSR) is a shift register whose input bit is a linear function of its previous state.

The only linear functions of single bits are xor and inverse-xor; thus it is a shift register whose input bit is driven by the exclusive-or (xor) of some bits of the overall shift register value.

The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle.

Linear Feedback Shift Registers sequence through  $(2^n - 1)$  states, where  $n$  is the number of registers in the LFSR. At each clock edge, the contents of the registers are shifted right by one position. There is feedback from predefined registers or taps to the left most register through an exclusive-NOR (XNOR).

## Chapter 2 Implementation



Top level module input and output signals

Signal		Function
clear	Optional	Synchronous clear, set output to zero
set	Optional	Synchronous set, set output to predefined value
rew	Optional	Count backwards, rewind
cke	Optional	Clock enable
clk	Mandatory	Clock signal
rst	Mandatory	Asynchronous reset
q	Optional	State of internal shift register
q_next	Optional	Next state of internal shift register
z	Optional	Zero flag
zq	Optional	Zero flag, clocked output
level1	Optional	Programmable level flag
level2	Optional	Programmable level flag

The counter is defined by updating a spreadsheet (vc.xls):

Name	type					
vcnt	LFSR					
clear	set	cke	rew			
	1	1	1	1		
q	q_bin	z	zq	level1	level2	
	1	0	0	0	0	0
wrap	wrap_around					
	0	1				
length	clear_value	set_value	wrap_value	level1	level2	
	4	0	1	0	15	

From these definitions the design is generated with two scripts. The first converts the excel file into a CSV file and the second script generates the verilog module based on the CSV file.

A Makefile, included in the project, can be used for this.

Makefile:

all:

```
    excel2csv vc.xls -S ,
    ./versatile_counter_generator.php Versatile_counter.csv > vc.v
```

Note:

The name of the generated CSV file is defined in the excel tab.

This design is done in Verilog RTL. The design is as a generic module.

The following options are present in the application specific define file:

parameters	Valid options	Comment
Length	2-32	Length of counter state vector
clear_value	Any counter value	Default value from excel sheet
set_value	Any counter value	Default value from excel sheet
wrap_value	Any counter value	Default value from excel sheet
level1_value	Any counter value	Default value from excel sheet
level2_value	Any counter value	Default value from excel sheet

**Note:**

For gray type counters output q\_next is next binary value

## Usage notes

1. All control signals are active high. To change this behavior use the following method for individual signals  
`.clear(~clear)`
2. Signal *clear* has priority over signal *set*.  
 The user can change this behavior with the following instantiation method  
`.clear(clear & ~set)`  
`.set(set)`
3. Both *CNT\_SET\_VALUE* and *CNT\_WRAP\_VALUE* are implemented as parameters and can be changed per instance with *defparam* Verilog statements
4. LEVEL1 and LEVEL2 should be used in implementations with REW function implemented. The levels can be changed per instance with *defparam* Verilog statements
5. To construct an up / down counter enable *CNT\_REW* and use inputs as below

	<b>rew</b>	<b>cke</b>
<b>up</b>	0	1
<b>down</b>	1	1
<b>-</b>	X	0

6. If *CNT\_REW* is enabled for a down counter wrap will occur when current state is CNT=0, next state will be *CNT\_WRAP\_VALUE*.

## Appendix A: excel2csv

---

This projects uses a script to convert excel spreadsheet(s) into comma separated value file(s) (csv files).

This program is available from CPAN:

<http://search.cpan.org/~leocharre/excel2csv-1.01/>

```
excel2csv - dump all worksheets from an Excel file to CSV files  
=====
```

The original script was written by Drew Broadley, with contributions from Aristotle Pagaltzis.

Packaged by Leo Charre leocharre at cpan dot org.

excel2csv takes any number of Excel files on the command line, reads them, and dumps each worksheet therein to a separate CSV file, named after the worksheet.

### INSTALLATION

```
perl Makefile.PL  
make install
```

### DEPENDENCIES

```
Getopt::Long  
Pod::Usage  
Spreadsheet::ParseExcel
```

### LICENCE

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.8 or, at your option, any later version of Perl 5 you may have available.



## Appendix B

### Versatile LFSR counter PHP scripts

---

To ease the usage of the LFSR type counter a script can be used. The script, VersatileCounter, is written in PHP and can be found in the SW directory.

The script takes two or three argument

1. length of LFSR polynom
2. number of increments
3. optional verbose flag

Example

To see state after 14 increments of a 4 bit counter

```
./VersatileCounter.php 4 14
```

```
1000
```

The verbose flag gives a more detailed output

```
./VersatileCounter.php 4 15 v
```

```
0: 0000
```

```
1: 0001
```

```
2: 0011
```

```
3: 0111
```

```
4: 1110
```

```
5: 1101
```

```
6: 1011
```

```
7: 0110
```

```
8: 1100
```

```
9: 1001
```

```
10: 0010
```

```
11: 0101
```

```
12: 1010
```

```
13: 0100
```

```
14: 1000
```

```
15: 0000
```

## Recommended Resources

---

**ORSoC** – <http://www.orsoc.se>

**ORSoC** is a fabless ASIC design & manufacturing services company, providing RTL to ASIC design services and silicon fabrication service. **ORSoC** are specialists building complex system based on the OpenRISC processor platform.

**Open Source IP** – <http://www.opencores.org>

Your number one source for open source IP and other FPGA/ASIC related information.

**XILINX** – <http://www.xilinx.com>

[http://www.xilinx.com/support/documentation/application\\_notes/xapp052.pdf](http://www.xilinx.com/support/documentation/application_notes/xapp052.pdf)