Versatile counter

A reconfigurable binary, gray or LFSR counter

Brought to You By ORSoC / OpenCores

Legal Notices and Disclaimers

Copyright Notice

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.

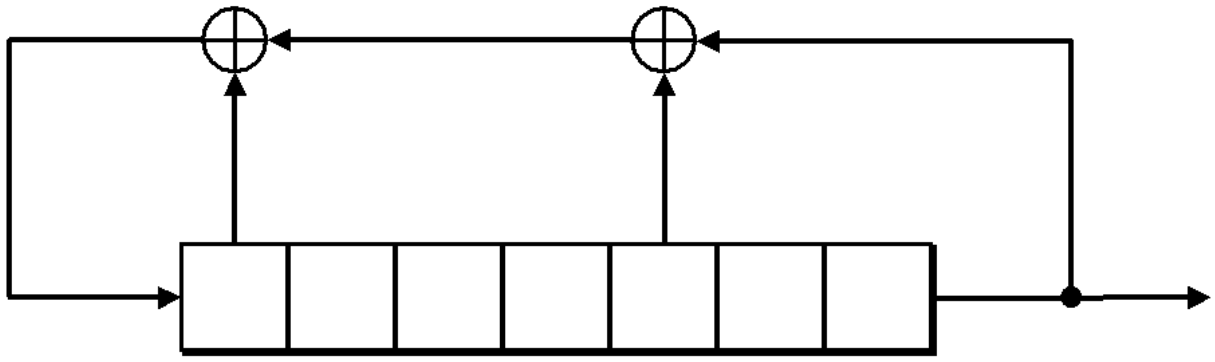
Distribution Rights

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

Chapter 1 Linear Feedback Shift Registers



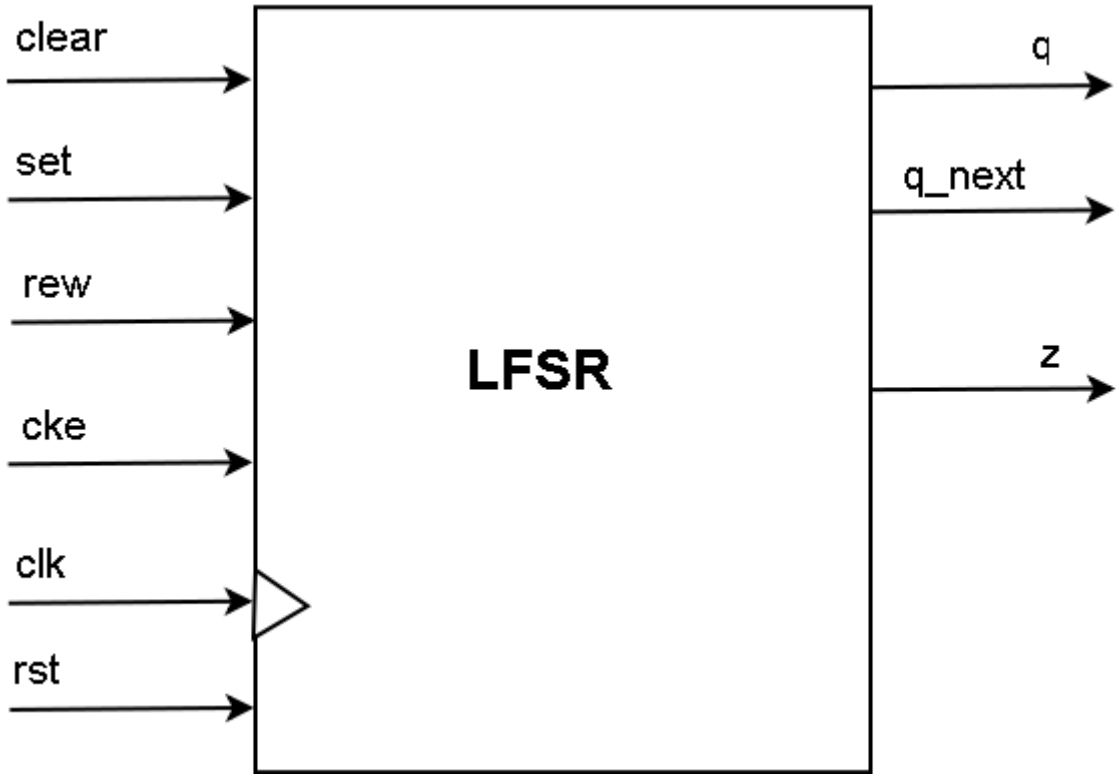
A **linear feedback shift register** (LFSR) is a shift register whose input bit is a linear function of its previous state.

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



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

This design is done in Verilog RTL. The design is as a generic module. Optional signals and other implementation details are controlled from a define file. This file in combination with the generic design generates the actual design file.

Two different standalone Verilog preprocessors is used in build of the actual design file. The preprocessor can be obtained as outlined below.

vpp:

1. in Debian / Ubuntu
   apt-get install vbpp
2. in windows / cygwin
   is part of ACTEL Libero

vppp:

http://search.cpan.org/~wsnyder/Verilog-Perl-3.045/vppp

Four files are used for the creation of the design

1. versatile_counter.v
   generic counter implementation
2. define file
   application specific define file with implementation options
3. lfsr_polynom.v
   File with definition of maximum length feedback polynomial
4. copyright.v
   OpenCores.org LGPL copyright notice

A Makefile in the rtl directory is used for the build of the actual output.

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

Define	Valid options	Comment
CNT_MODULE_NAME		Sets module name of design
CNT_TYPE	BINARY, GRAY, LFSR	Defines count sequence
CNT_LENGTH	2-32	Defines number of taps
CNT_CLEAR		If defined activates signal clear
CNT_SET		If defined activate signal set
CNT_SET_VALUE		Value to apply to CNT vector
CNT_REW		If defined activates signal rew. If input is active counter direction is changed
CNT_CE		If define activates signal ce, clock enable
CNT_WRAP		Define to generate a shorter cycle than maximum
CNT_WRAP_VALUE		Defines last state CNT vector prior to wrap to zero
CNT_Q		If defined q is available as output
CNT_Q_BIN		For gray type counters optional binary output
CNT_QNEXT		If defined q_next available as output
CNT_Z		If defined activates a signal indicating state of CNT is zero
CNT_ZQ		For optimal timing performance output zq is a register value based on q_next

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. Signal *clear* and *set* are normally not affected by optional clock enable. To change this behavior use the following method
   .set(set & cke)
4. Both *CNT_SET_VALUE* and *CNT_WRAP_VALUE* are implemented as parameters and can be changed per instance with *defparam* Verilog statements
5. To construct an up / down counter enable *CNT_REW* and use inputs as below

	rew	**cke**
up	0	1
down	1	1
-	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*.

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