An Introduction to Ring Counters

Jesse op den Brouw

The Hague University of Applied Sciences

February 10, 2016

Abstract

Ring counters are a type of counters composed of shift registers. They are extremely fast and require few logic gates. This document introduces two types of ring counters, the straight ring counter and the Johnson counter. The latter is elaborated on in detail. Some applications are described as well as commercially available devices and designs using Johnson counters. A generic VHDL description and a description of the commercially available 4017 are provided.

Table of Contents

List of Figures

List of Listings

Any questions or comments regarding this document can be sent to J.E.J.opdenBrouw@hhs.nl

1 Ring Counters

Ring counters are a type of counter created using *shift registers*. A shift register is constructed using D-type flip-flops where the output of one flip-flop is connected to the input of another flip-flop [1]. With ring counters, the output of the last flip-flop is fed to the input of the first flip-flop. Ring counters do not count using normal binary code, but their internal state can be used to decode to any output sequence wanted.

There are two types of ring counters:

a) Straight ring counter

b) Johnson counter

There is a third counter type using a shift register, the Linear Feedback Shift Register (LFSR). It has a more elaborate feedback circuit than the other two ring counters and is therefore not discussed in this document.

2 Straight ring counter

A straight ring counter or Overbeck counter connects the output of the last flip-flop to the first flip-flop input and circulates a single one bit around the ring. It provides a *one-hot* counting sequence. For example, in a 4-register ring counter, with initial register values of 1000, the repeating sequence is 1000, 0100, 0010, 0001. Note that one of the flip-flops must be pre-loaded with a logic 1 in order for it to operate properly. Also note that an n-bit ring counter cycles through exactly n states.

A schematic of a 4-bit straight ring counter is given in Figure 1. The asynchronous reset will set the initial contents of the counter to 1000 (note that the least significant flip-flop is on the left side).

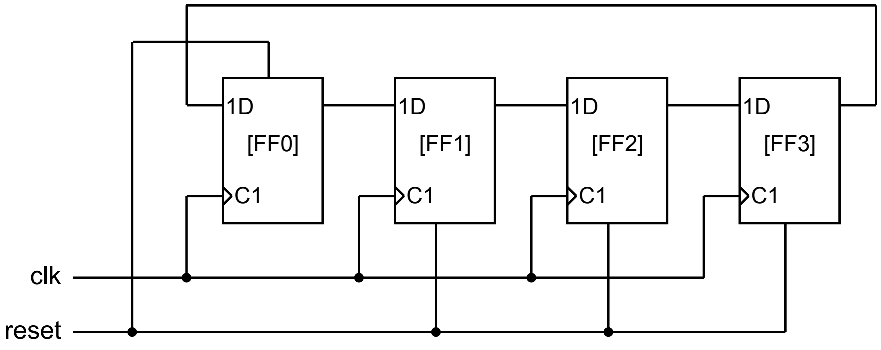

Figure 1: *A 4-bit straight ring counter.*

Straight ring counters are currently seldom used and there are no commercial devices available. In the past, they were mainly used as decimal counters using *Nixie tubes* and *neon tubes* [2, 3].

3 Johnson Counter

Another form of ring counter is created by feeding back the complement of the contents of the last flip-flop to the input of the first flip-flop. This is called a *twisted ring counter*, but is better known as the *Johnson counter*. The alternative term Möbius counter is found in many books and articles because the Johnson counter resembles the famous Möbius strip [4]. For example, in a 5-flip-flop Johnson counter with an initial register contents (or state) of 00000, the repeating sequence is 00000, 10000, 11000, 11100, 11110, 11111, 01111, 00111, 00011, 00001. When observing the pattern, it can be seen that any changes between succeeding states, only one flip-flop changes state. As a result, any of these states is directly, spike-free decodable with only a two-input gate [5, 6].

Figure 2 shows a 5-bit Johnson counter with *terminal count* output `tc`. This output becomes high in the last counting state 00001 before returning to 00000.

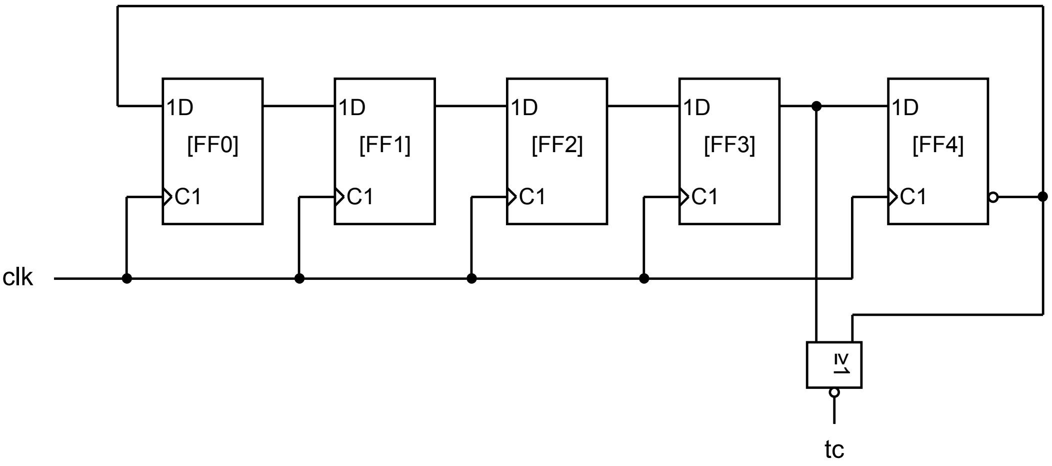

Figure 2: *A 5-bit Johnson Counter with Terminal Count. The reset is omitted for clarity.*

Only the two most significant flop-flops have to be sampled, so *tc* is logic '1' when Q_3 is logic '0' and Q_4 is logic '1'. Using De Morgan's theorem, the resulting function can be realised using a NOR gate and the already available flip-flop outputs.

$$tc = \overline{Q_3} \cdot Q_4 = \overline{Q_3 + \overline{Q_4}} \tag{1}$$

It can be easily shown that an n-bit Johnson counter cycles through exactly $2n$ states. This becomes increasingly inefficient for $n > 2$ since any n-bit counter has 2^n states by itself. If the Johnson counter enters a state that does not belong to the original counting sequence, it will exhibit a single *parasitic counting sequence* with $2^n - 2n$ states [7, 8]. This is why most commercial devices incorporate *self-correcting* logic. Within a full counting sequence or so, the device will resume normal operation, as shown for example in the schematic in [9].

Johnson counters are very fast because there is no logic between the output of one flip-flop and the input of another flip-flop, i.e., there is no next state logic (one would expect an inverter in the feedback loop but the inverted output of a flip-flop is usually available, especially on ASIC's [10]). Therefore the maximum frequency at which the counter operates reliably is given by (2).

$$f_{max} = \frac{1}{t_{P(max)}(\mathrm{FF}) + t_{su}(\mathrm{FF})} \tag{2}$$

where f_{max} is the maximum obtainable frequency, $t_{P(max)}(\mathrm{FF})$ is the maximum propagation delay of the flip-flop outputs with respect to the active clock edge and $t_{su}(\mathrm{FF})$ is the setup time of the flip-flops. The propagation delay of output `tc` with respect to the active clock edge is given by (3).

$$\begin{aligned} t_{P(max)}(\mathrm{tc}) &= t_{P(max)}(\mathrm{FF}) + t_{P(max)}(\mathrm{NOR}) \\ t_{P(min)}(\mathrm{tc}) &= t_{P(min)}(\mathrm{FF}) + t_{P(min)}(\mathrm{NOR}) \end{aligned} \tag{3}$$

where $t_{P(min)}$ and $t_{P(max)}$ are the minimum and maximum propagation delays of the named signals.

4 Applications of Johnson counters

A 5-stage Johnson counter can be used as a decade frequency divider [11].

Johnson counters are used as multiphase clock signal generators. Such generators are used in large digital systems where timing signals are needed with high accuracy with respect to the central clock [12]. Johnson counters produce glitch free symmetric (50% duty cycle) outputs per count cycle [13]. A good example is given by Harris [14].

Amann et al. [15] have shown that designing FSMs using loadable Johnson counters as state memory saves up to 30% of the product terms generating the next state and outputs.

Johnson counters are used to drive stepper motor circuits [16, 17].

5 Commercially available Johnson counters

The CD4017B is an all-CMOS, commercially available 5-stage Johnson counter with decoded outputs [9]. It is used in a wide variety of designs, mostly as a 10-stage counter (counting to 10 is done as long as mankind exists). The device is very popular and cheap but it has a big design flaw: clock gating. The `clock` signal is AND-gated with the `clock_inhibit` signal. It produces pulse-triggered behaviour of the clock input (`clock_inhibit` must be held stable when `clock` is high) and exhibits clock skew (the resulting clock pulse is delayed by the AND gate). Note that the internal shift register contents is not available on output pins, only the ten decoded outputs are available making the 4017 behave as a straight ring counter.

The CD4022B is essentially a 4-stage version of the 4017 [18].

The 70HC/HCT4017 is a 7400 remake of the 4017. The HCT has TTL compatible power supply, inputs and outputs.

6 Design using Johnson Counters

A very nice design using Johnson counters is the LFO generator by R.G. Keen [19].

A number of 4017 designs, including an automatic bathroom light switch and a led chaser circuit, can be found on [20].

Using a number of 4017, one can build a Nixie tube digital clock [21].

The 4026 and 4033 Decade Counters consist of a 5-stage Johnson counter similar to the 4017 and an output decoder that converts the counter state to 7-segment decoded outputs [22].

Ben Kuiper describes a range detection circuit using 4017's (and more 4000-series devices) in [23].

7 VHDL description of a Johnson counter

Describing a Johnson counter in VHDL is easy and straightforward. Using the schematic of Figure 2, the entity in Listing 1 consists of a clock input, an asynchronous reset input, a count output and a terminal count output.

The counter is set to all zeros when the reset input is high. The rising edge is chosen as active clock edge. On the rising edge of the clock the contents of the register is shifted one stage towards the most significant flip-flop and the least significant flip-flop is loaded with the complement of the most significant flip-flop. This is done using standard VHDL implementation with the & concatenation operator.

This implementation makes use of so-called *generic constants*. The design is written as an n-stage Johnson counter. During instantiation, the generic constant n has to be supplied as a positive number. When absent, the design is instantiated with $n = 5$.

```vhdl
-- Filename:     johnson_counter.vhd
-- Filetype:     VHDL Design Unit
-- Date:         5 mar 2015
-- Update:       -
-- Description:  VHDL Description of a n-stage Johnson Counter
-- Author:       J. op den Brouw
-- State:        Demo
-- Error:        -
-- Version:      1.0
-- Copyright:    (c)2015, De Haagse Hogeschool

-- This VHDL description describes the behaviour of n-stage
-- Johnson counter with spike free active high decoded terminal
-- count output.

-- Using the 1164 std_logic
library ieee;
use ieee.std_logic_1164.all;

-- Tbe port description of the Johnson counter
entity johnson_counter is
```

```
    generic (n : integer := 5);
    port (clk :    in std_logic;
          areset : in std_logic;
          count :  out std_logic_vector (0 to n-1);
          tc :     out std_logic
         );
end entity johnson_counter;

architecture rtl of johnson_counter is
-- Internal counter signal
signal count_int : std_logic_vector (0 to n-1);
begin

    -- The Johnson counter itself
    process (clk, areset) is
    begin
        -- The reset is active high
        if areset = '1' then
            -- Set all counter bits to 0, nice VHDL trick
            count_int <= (others => '0');
        elsif rising_edge(clk) then
            -- Shift the lot a stage and feed back the last one
            count_int <= not count_int(n-1) & count_int(0 to n-2);
        end if;
    end process;

    -- The outputs
    count <= count_int;
    -- tc high when counter is ...01, where the dots should be all zeros.
    -- tc <= (not count_int(n-1)) nor count_int(n-2);
    -- tc <= '1' when count_int(n-1) = '1' and count_int(n-2) = '0' else '0';
    tc <= '1' when count_int(n-2 to n-1) = ”01” else '0';

end architecture rtl;
```

Listing 1: *A VHDL description of an n-stage Johnson counter*

8 The 4017 in VHDL

In Listing 2, a VHDL description of the 4017 Johnson counter is presented. The description is created using the schematic in [9]. There is one omission: in the original design, the clock input has a Schmitt-trigger input.

```
-- Filename:     cd4017b.vhd
-- Filetype:     VHDL Design Unit
-- Date:         5 mar 2015
-- Update:       -
-- Description:  VHDL Description of a CD4017B 5-stage Johnson Counter
-- Author:       J. op den Brouw
-- State:        Demo
-- Error:        -
-- Version:      1.0
-- Copyright:    (c)2015, De Haagse Hogeschool
--
```

```
-- This VHDL description describes the behaviour of an CD4017B 5-stage
-- Johnson counter with spike free active high decoded outputs.
-- See http://www.ti.com/lit/ds/symlink/cd4017b.pdf for more information.
--
-- In some designs, the naming of the pins is different:
--      clock -> CP0
--      clock_inhibit -> ~CP1   or    not(CP1)
--      reset -> MR
--      carry out -> not(Q)5--9
--
-- Please note that this design uses clock-gating: the clock is gated
-- with an AND gate and a signal clock_inhibit. This is a poor hardware
-- design. It produces both pulse-triggered behaviour of the clock input
-- (clock_inhibit must be held stable when clock is high) and clock skew
-- (the resulting clock pulse is delayed by the AND gate).
-- PLEASE USE THIS DESIGN FOR REFERENCE ONLY, NOT IN REAL PRODUCTS.

-- Using the 1164 std_logic
library ieee;
use ieee.std_logic_1164.all;

-- The port description of the 4017
entity cd4017b is
    port (clock :         in std_logic;
          reset :         in std_logic;
          clock_inhibit : in std_logic;
          q :             out std_logic_vector (9 downto 0);
          carry_out :     out std_logic
         );
end entity cd4017b;

architecture gate_level of cd4017b is
-- In the datasheet, FF1 is on the left side
signal count_int : std_logic_vector (1 to 5);
-- Internal signal for clock gating
signal clk_int : std_logic;
begin
    -- The clock is gated
    clk_int <= (not clock) nor (not (not clock_inhibit));

    -- The Johnson counter itself
    process (clk_int, reset) is
    begin
        if reset = '1' then
            count_int <= "00000";
        elsif rising_edge(clk_int) then
            count_int(5) <= count_int(4);
            count_int(4) <= count_int(3);
            count_int(3) <= not (((not count_int(1)) and (not count_int(3))) or
                                  (not count_int(2)));
            count_int(2) <= count_int(1);
            count_int(1) <= not count_int(5);
        end if;
     end process;

    -- The outputs
    carry_out <= not count_int(5);
    q(0) <= not (not count_int(1) nand not count_int(5));
    q(1) <= not (    count_int(1) nand not count_int(2));
    q(2) <= not (    count_int(2) nand not count_int(3));
    q(3) <= not (    count_int(3) nand not count_int(4));
    q(4) <= not (    count_int(4) nand not count_int(5));
    q(5) <= not (    count_int(1) nand     count_int(5));
    q(6) <= not (not count_int(1) nand     count_int(2));
    q(7) <= not (not count_int(2) nand     count_int(3));
    q(8) <= not (not count_int(3) nand     count_int(4));
    q(9) <= not (not count_int(4) nand     count_int(5));

end architecture gate_level;
```

Listing 2: *A VHDL description of the 4017 5-stage Johnson counter*

References

[1] M. Morris Mano and M.D. Ciletti. *Digital Design*. Prentice Hall, 2006. ISBN: 978-0131989245 (cit. on p. 3).

[2] J.C. Manley and E.F. Buckley. “Ring Counter”. In: *Electronics* EC-14.5 (Jan. 1950), pp. 84–87 (cit. on p. 3).

[3] Ronald Dekker. *Ring Counter Variations*. URL: http://www.dos4ever.com/ring/ring.html (visited on 2/27/2015) (cit. on p. 3).

[4] Alexander Bogomolny. *Möbius Strip*. URL: http://www.cut-the-knot.org/do_you_know/moebius.shtml (visited on 3/5/2015) (cit. on p. 4).

[5] I. Brcic. ““Ideally Fast” Decimal Counters with Bistables”. In: *Electronic Computers, IEEE Transactions on* EC-14.5 (Oct. 1965), pp. 733–737 (cit. on p. 4).

[6] Nir Dahan. *The Johnson Counter*. URL: https://asicdigitaldesign.wordpress.com/2007/08/20/the-johnson-counter/ (visited on 2/21/2015) (cit. on p. 4).

[7] Mark Zwoliński. *Digital System Design with VHDL*. 2nd ed. Pearson, 2004. ISBN: 0-13-039985-X (cit. on p. 4).

[8] P. Wilson. *Design Recipes for FPGAs: Using Verilog and VHDL*. Elsevier Science, 2011, p. 235. ISBN: 9780080548425 (cit. on p. 4).

[9] Texas Instruments. *Datasheet CD4017B*. Feb. 2004. URL: http://www.ti.com/lit/ds/symlink/cd4017b.pdf (visited on 2/21/2015) (cit. on pp. 4, 5, 7).

[10] Neil H. E. Weste and David Money Harris. *CMOS VLSI Design, A Circuits and Systems Perspective*. 4th ed. Addison-Wesley, 2011, pp. 393–394. ISBN: 978-0-321-54774-3 (cit. on p. 5).

[11] R.M. Marston. *Modern CMOS Circuits Manual*. Newnes, 1996, pp. 178–179. ISBN: 0-7506-2565-1 (cit. on p. 5).

[12] A.P. Thijssen and H.A. Vink. *Digitale Techniek, van probleemstelling tot realisatie deel 2*. Dutch. 5th ed. Delft University Press, 2000. ISBN: 90-407-1838-5 (cit. on p. 5).

[13] Ben Cohen. *Real Chip Design and Verification Using Verilog and VHDL*. Vhdl-Cohen Publishing, 2002. ISBN: 0-9705394-2-8 (cit. on p. 5).

[14] William A. Harris. *Precision phase generator*. Patent WO 02/054598 A2. July 2002. URL: http://worldwide.espacenet.com/publicationDetails/originalDocument?FT=D&date=20020711&DB=&locale=en_EP&CC=WO&NR=02054598A2&KC=A2&ND=4 (visited on 2/22/2015) (cit. on p. 5).

[15] R. Amann, B. Eschermann, and U.G. Baitinger. "PLA based finite state machines using Johnson counters as state memories". In: *Computer Design: VLSI in Computers and Processors, 1988. ICCD '88., Proceedings of the 1988 IEEE International Conference on*. Oct. 1988, pp. 267–270 (cit. on p. 5).

[16] F.W. Lin. *Stepping motor drive circuit*. US Patent 3,893,012. July 1975. URL: http://worldwide.espacenet.com/publicationDetails/biblio?DB=EPODOC&II=0&ND=1&at=7&locale=en_EP&FT=D&CC=US&NR=3893012A&KC=A (cit. on p. 5).

[17] All About Circuits. *Ring Counter*. URL: http://www.allaboutcircuits.com/vol_4/chpt_12/6.html (visited on 2/28/2015) (cit. on p. 5).

[18] Texas Instruments. *Datasheet CD4022B*. Feb. 2004. URL: http://www.ti.com/lit/ds/symlink/cd4022b.pdf (visited on 2/21/2015) (cit. on p. 5).

[19] R.G. Keen. *Digital Generation of LFO's for Modulating Effects*. URL: http://www.geofex.com/article_folders/lfos/psuedorandom.htm (visited on 2/22/2015) (cit. on p. 6).

[20] *16 x IC 4017 Circuits and Projects*. URL: http://www.electroschematics.com/tag/4017-circuits/ (visited on 2/22/2015) (cit. on p. 6).

[21] Mike Harrison. *Build a Nixie Tube Digital Clock*. URL: http://www.electricstuff.co.uk/nixclock.html (visited on 2/22/2015) (cit. on p. 6).

[22] Texas Instruments. *Datasheet CD4026B and CD4033B*. 2003. URL: http://www.ti.com/lit/ds/symlink/cd4026b.pdf (visited on 2/22/2015) (cit. on p. 6).

[23] B. Kuiper. *Voorbeeldcircuit afstandsdetectie*. Dutch. Internal Report. Version 1.1. The Hague University of Applied Sciences, 2015 (cit. on p. 6).