Bergische Universität Wuppertal

Allgemeine Elektrotechnik und Theoretische Nachrichtentechnik

Design and Implementation of Digital Systems

Summer semester 2005

Associated Professor, Dr. Alexander A. Ivaniuk

Contact information:
University of Wuppertal
Communication Theory Department
Campus Freudenberg
Rainer Gruenter Straße, 21

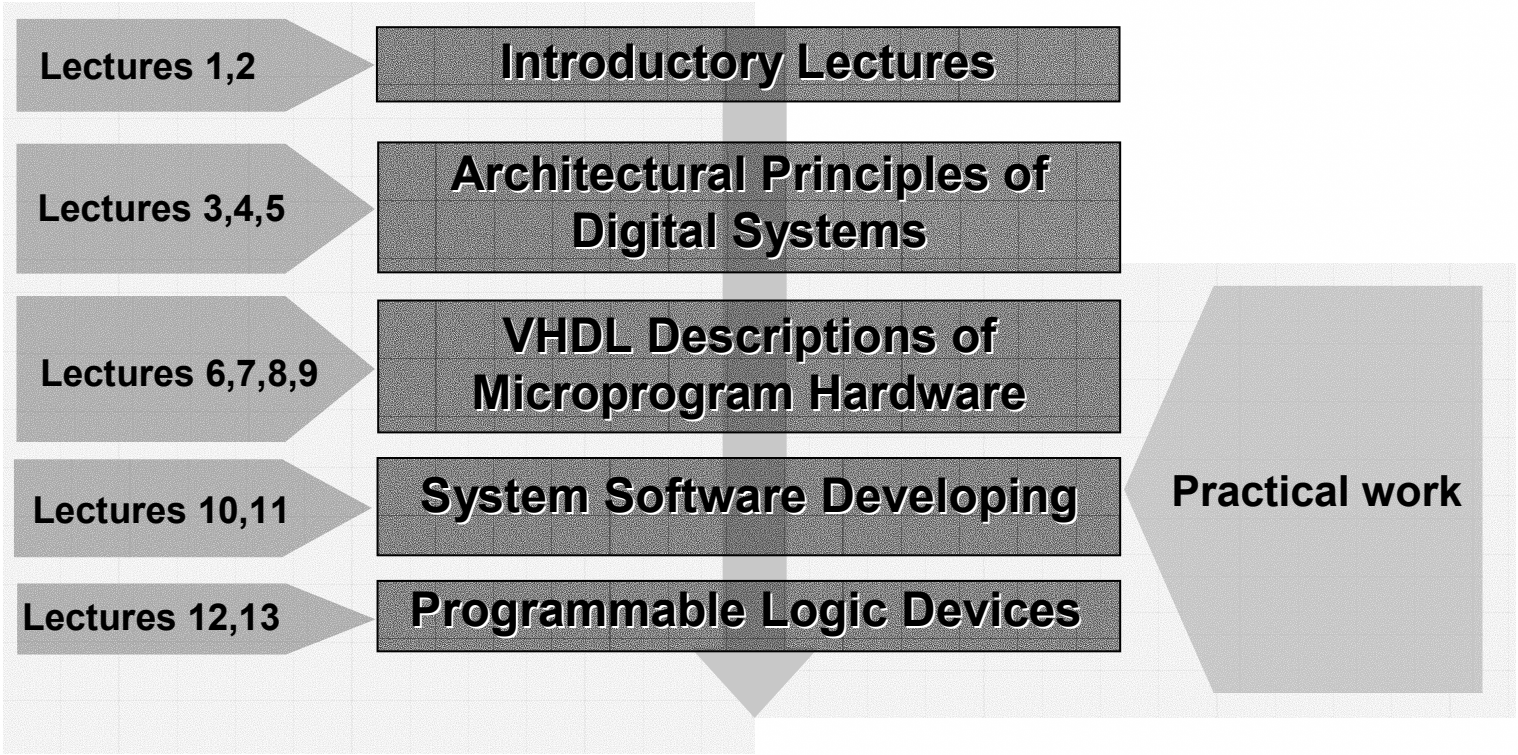
e-mail: ivaniuk@uni-wuppertal.de
phone : +49 (202) 439-1832

Scientific Activities: Random Access Memory Reliability Techniques:

- On-line and Off-line concurrent checking;
- Transparent diagnostic algorithms;
- Built-In Self-Diagnosis and Self-Repair Approaches.

SUBJECT MATTER : TOPICS, LECTURES AND PRACTICAL WORK

This course covers next main topics of **Advanced Digital Design.** The relation between topics, lectures and practical work is shown below.



This advanced course in **Digital Design** is more **practical** than theoretical one.

The main goal of practical work is to know how to implement **application specific hardware** for the **programmable logic devices** like **FPGA**. The practical work covers next topics:

1. VHDL description, simulation, implementation, and verification stages.
2. VHDL design and implementation of Finite State Machines.
3. The implementation of the various peripheral controllers.
4. The VHDL design of microprogram FSM architectures.
5. System software developing for microprogram hardware structures.
6. The implementation of application specific microprogram hardware.
7. Simulation, verification and testing of digital systems.

Each task for the practical work has **credit points**. Student can perform at least only one task, but it is up to him to reach more credits.

Students should organize the **groups** by 2 at least (more students – more complicated task).

The semester will ends with **small project** by application of the specific hardware on the base of DIGILENT SPARTAN-3 prototyping board.

Here is a list of software and hardware tools for the practical work:

1. **FPGA Advantage IDE** (www.mentor.com):
   - Design Creation;
   - Simulation (ModelSim);
   - Synthesis (Precision Synthesis / Leonardo Spectrum).
2. **Xilinx ISE** (www.xilinx.com):
   - Design Creation;
   - Simulation (ModelSim);
   - Synthesis RTL / Physical;
   - Design Implementation (translation, mapping, place & route);
   - Device Configuration (programming);
   - Post implementation verification (ChipScope).
3. **C/C++ compiler.**
4. **DIGILENT Spartan-3 prototyping board** (www.digilentinc.com).

Prescaler/Divider - digital device which brings down the high input digital frequency of discrete signal.

Johnson counter - linear shift register with one negative feedback, a.k.a. twisted ring counter, or Moebuis counter.

INTRODUCTION

- Digital frequency dividers and prescalers
- Special shift registers
- Glitches and hazards in digital circuits
- Useful materials and links

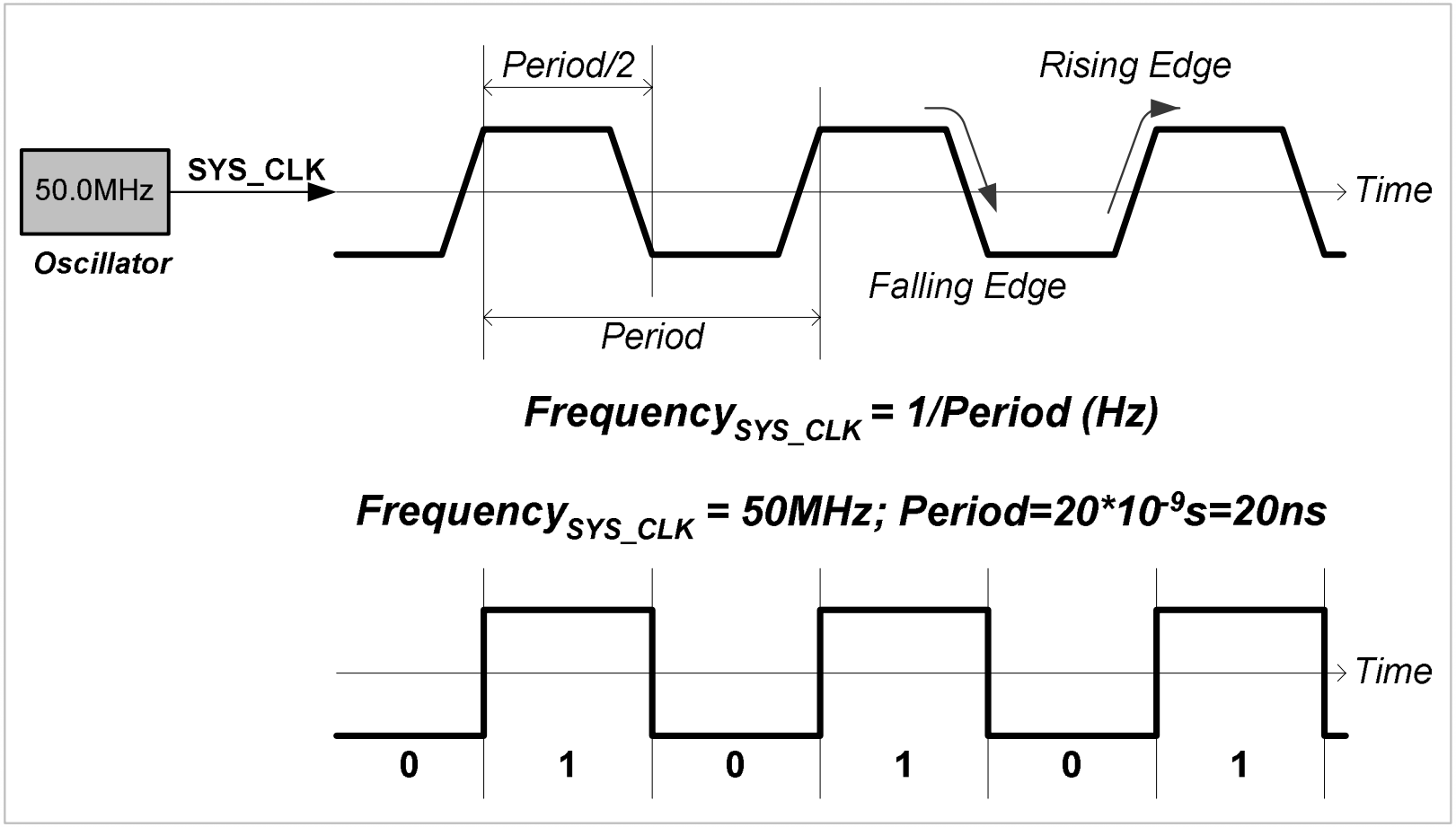
Digital frequency dividers and prescalers: they are digital circuits which bring down the high frequency input signal. Usually when we are talking about digital frequency we are thinking about digital clock signals. They are produced by oscillators (quartz resonators). The clock signals are in use for several purposes:

1. **Correct functionality of sequential components**:
   - flip-flops, registers, shift registers, counters, FSMs, etc.
2. **Digital system components synchronization**:
   - synchronization of the uP pipe-line stages.
3. **Synchronous data exchange interfaces**:
   - interfaces to the synchronous memories, synchronous serial and parallel interfaces as JTAG (Boundary Scan), I2C, SPI/Microwire, etc.

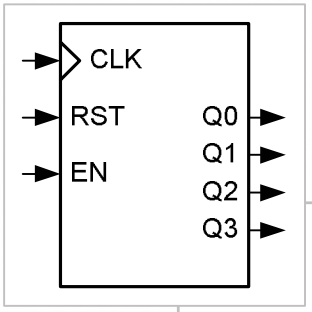
Usually one digital system has one (global) clock signal for synchronization. It calls SYSTEM CLOCK or SYSTEM FREQUENCY. Internally digital system may has several clock signals (several frequencies): system bus frequency, external interface frequency, etc.

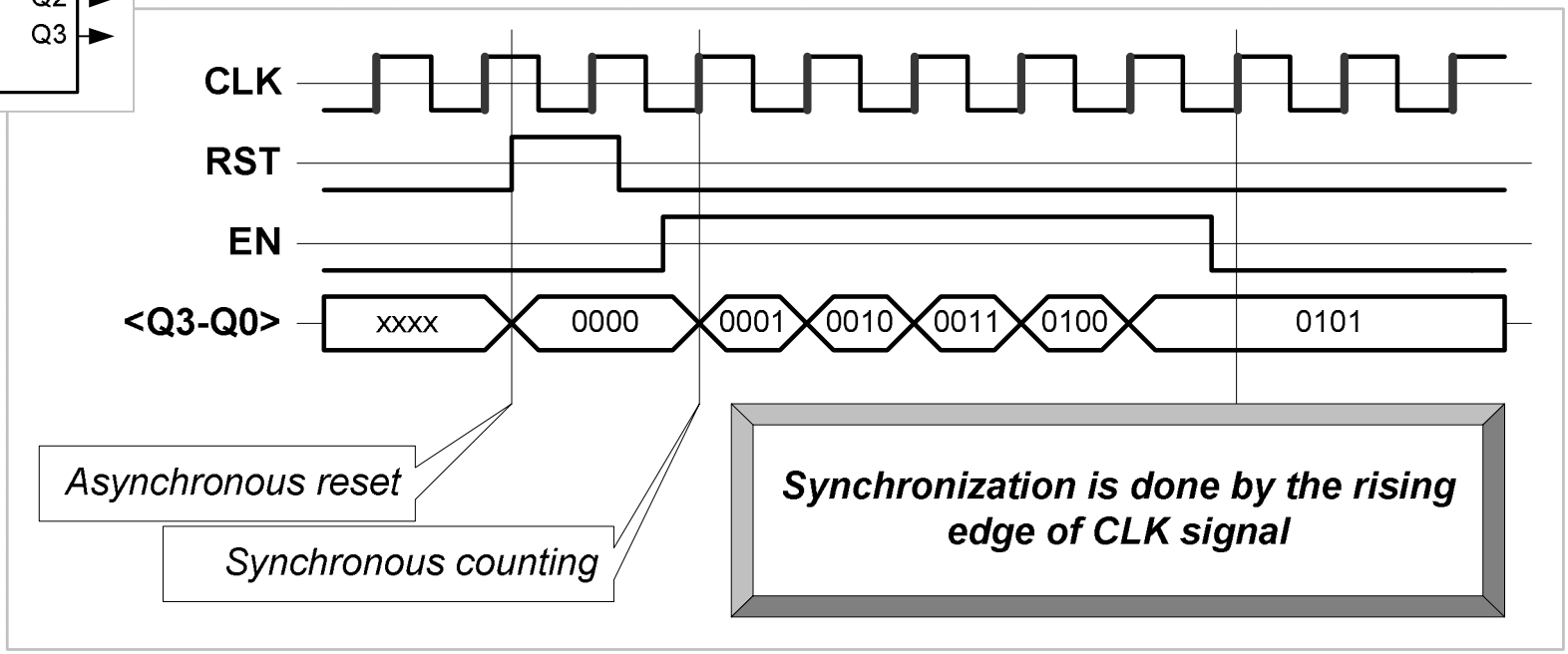
***NOTE:* Intel Pentium 4 Extreme Edition has System Clock frequency up to 3.73 GHz**

Digital frequency dividers and prescalers. The clock signal generated by oscillator has next shape and properties:



Digital frequency dividers and prescalers. Digital circuits which functionality depends on system clock events (rising or/and falling edges) named as SYNCHRONOUS circuits. For example synchronous binary "up" counter with asynchronous reset and enable has next symbol and ports timing diagram:





Digital frequency dividers and prescalers. The VHDL descriptions of such sequential circuits as binary counters have synchronous and asynchronous blocks and processes:

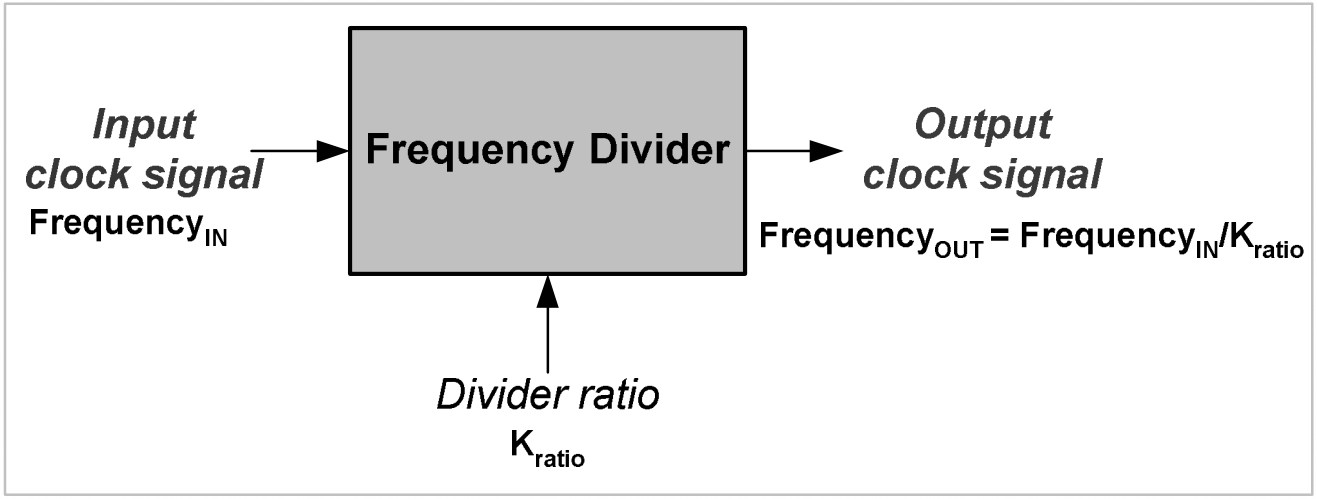
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

ENTITY counter4 IS
   port(
      CLK,RST,EN: in std_logic;
      Q: out std_logic_vector(3 downto 0)
   );
END counter4;

--
ARCHITECTURE beh OF counter4 IS
   signal counter: std_logic_vector(3 downto 0);
BEGIN
   MAIN: process(CLK,RST,EN,counter)
   begin
      if RST='1' then
         counter<=(others=>'0');
      elsif CLK'event and CLK='1' then
         if EN='1' then
            counter<=counter+1;
         end if;
      end if;
   end process;
   Q<=counter;
END beh;
```

Asynchronous reset condition

Synchronous counting

Asynchronous connection to the output port

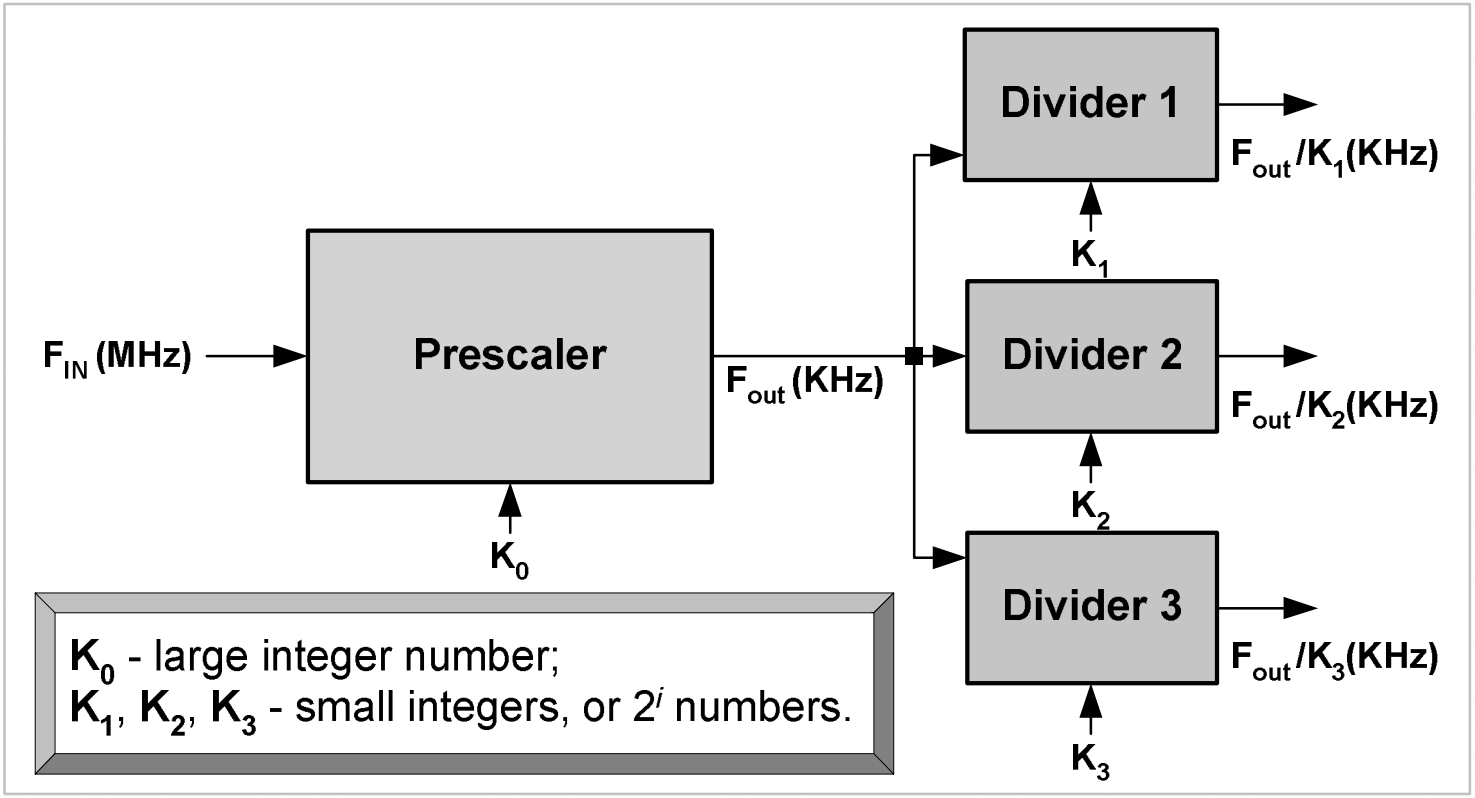
Digital frequency dividers and prescalers. Frequency dividers and prescalers generate the output clock signals which have the frequency smaller that the input clock signal. The divider circuit has a property as the DIVIDER RATIO.



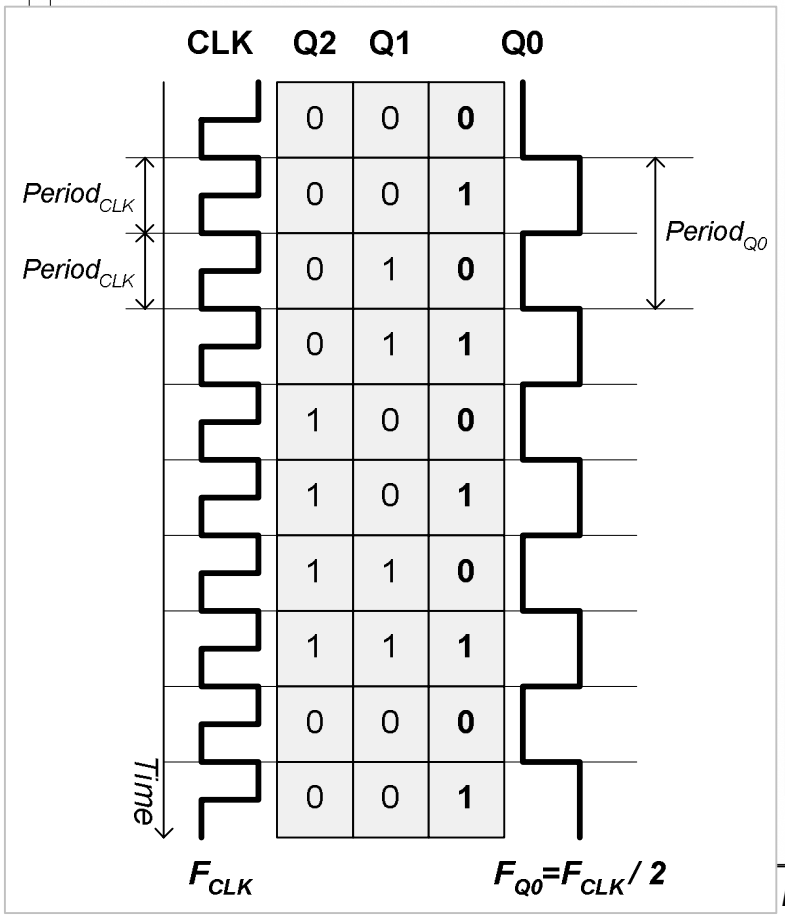
We will consider three variants of K_{ratio} values:

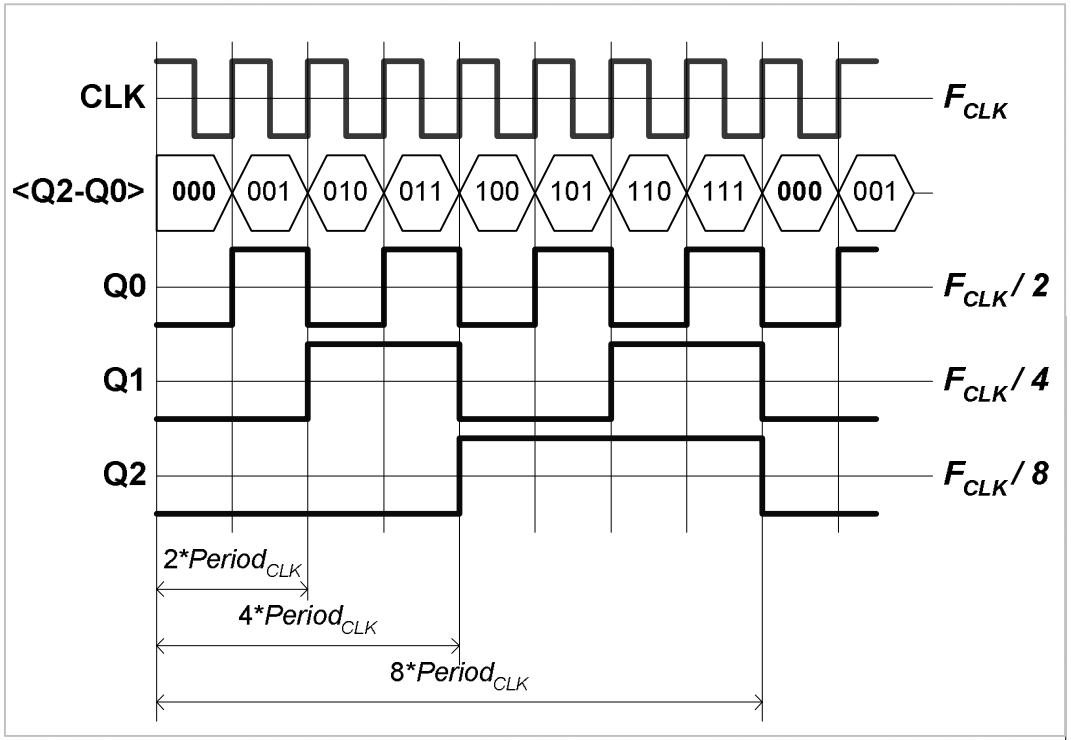
1. $K_{ratio}=2^i$, where $i=1,2,3,..,N$: 1,2,4,8,16,32,…
2. $K_{ratio}=i$, where i – is a small integer : 1,2,3,4,5,6,…
3. $K_{ratio}=2*i$, where i – is a large integer: 50’000’000,…

Digital frequency dividers and prescalers. If there are several frequency dividers on the same chain, than the first divider, which is the source for other dividers, is called PRESCALER.

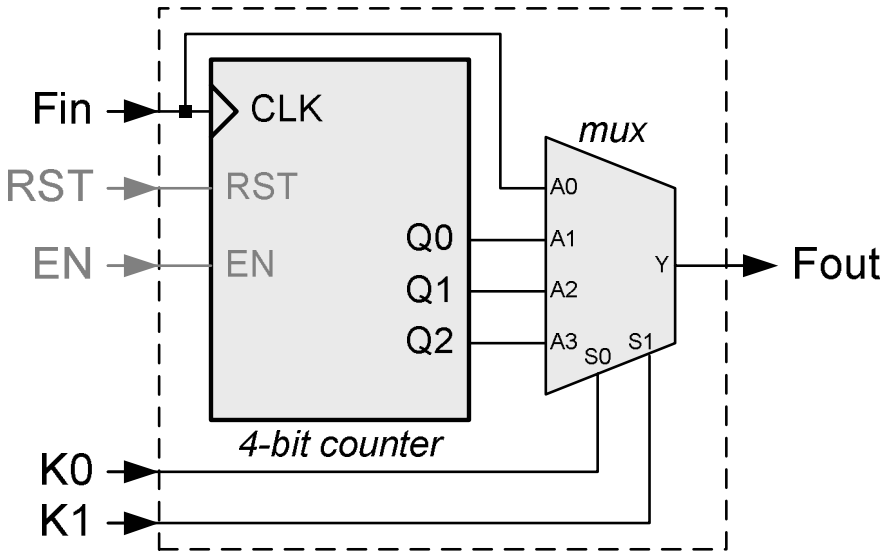


Digital frequency dividers and prescalers. Let us exam the behavior and structure of the simplest frequency divider with $K_{ratio}=2^i$. The binary "up" synchronous counter can be used as such frequency divider. If the functionality of a M-bit counter depends on the rising edge event of the incoming clock signal with frequency F_{IN} than the outgoing frequency from the j bit of the counter equals to $F_{OUT(j)}=F_{IN}/2^j$, where $j=1,2,\ldots,M$.





Digital frequency dividers and prescalers. To build the configurable frequency divider next schematic may be used:

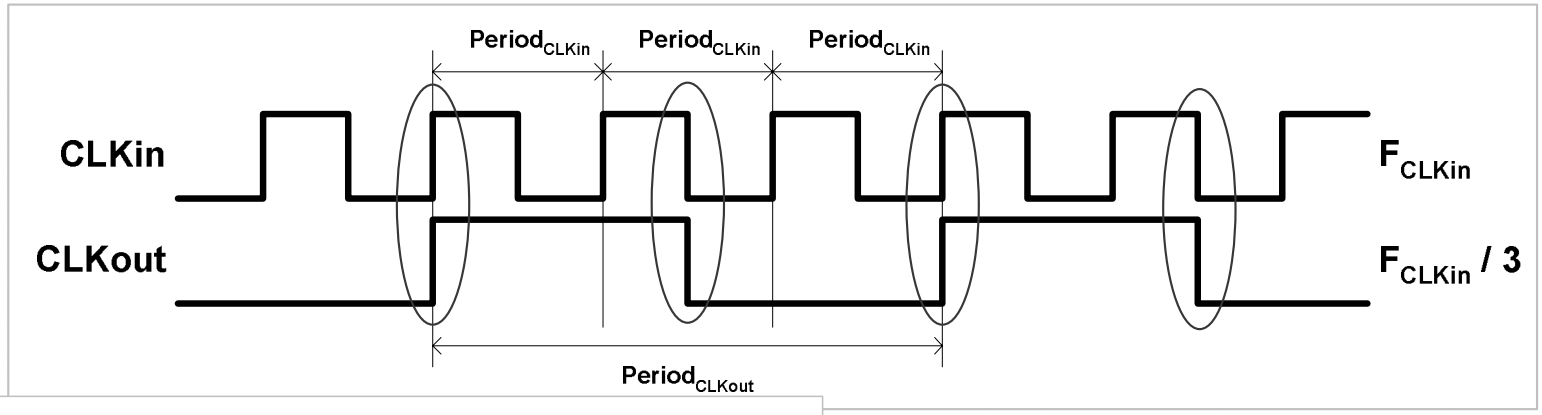


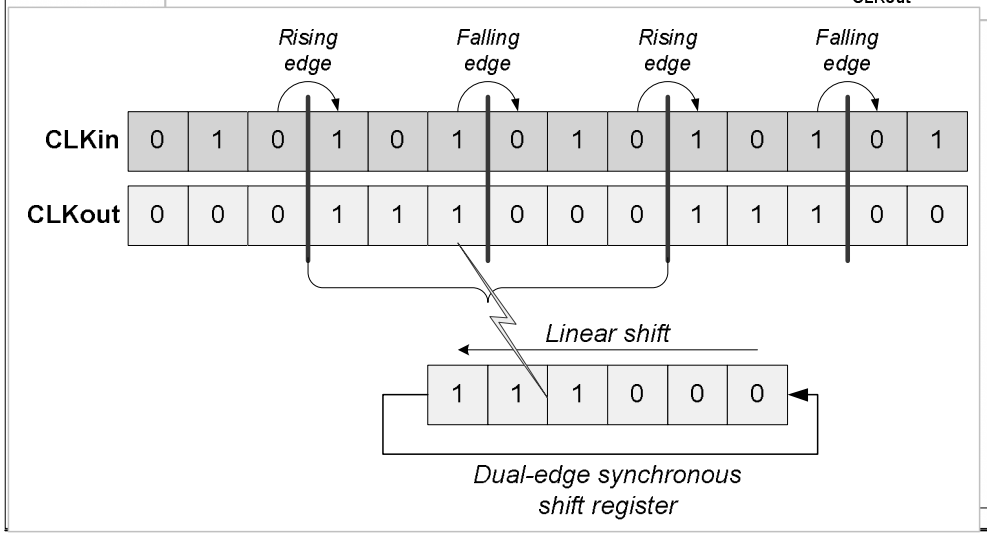

```
ENTITY divider3 IS
   port(
      Fin,RST: in std_logic;
      K: in std_logic_vector(1 downto 0);
      Fout: out std_logic
   );
END divider3;
```

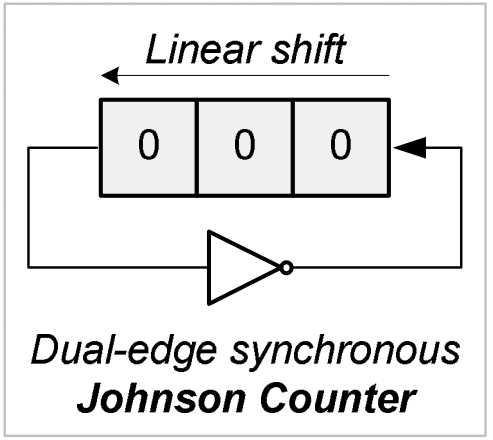
K1	K0	K_{ratio}
0	0	1
0	1	2
1	0	4
1	1	8

```
--
ARCHITECTURE beh OF divider3 IS
   signal Q: std_logic_vector(2 downto 0);
   signal org: std_logic;
   signal Y: std_logic;
BEGIN
   org<=Fin;

   COUNTER: process(Fin,RST,Q)
   begin
      if RST='1' then
         Q<=(others=>'0');
      elsif Fin'event and Fin='1' then
         Q<=Q+1;
      end if;
   end process;

   MUX: process(K,Q,org)
   variable A:std_logic_vector(3 downto 0);
   begin
      A:=Q & org;
      Y<=A(CONV_INTEGER(K));
   end process;

   Fout<=Y;
END beh;
```

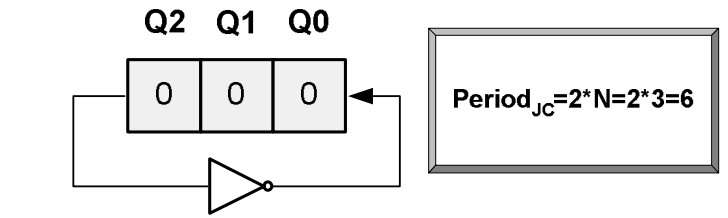
Digital frequency dividers and prescalers. Next types of frequency dividers are dividers with $K_{ratio}=i$, where i is a small integer. We can design the divider with $K_{ratio}=2$ and $K_{ratio}=4$. And what about divider with $K_{ratio}=3$ or $K_{ratio}=5$? Let us have a look to the timing diagram of the outgoing clock signal which has a frequency $F_{out}=F_{in}$ / 3.



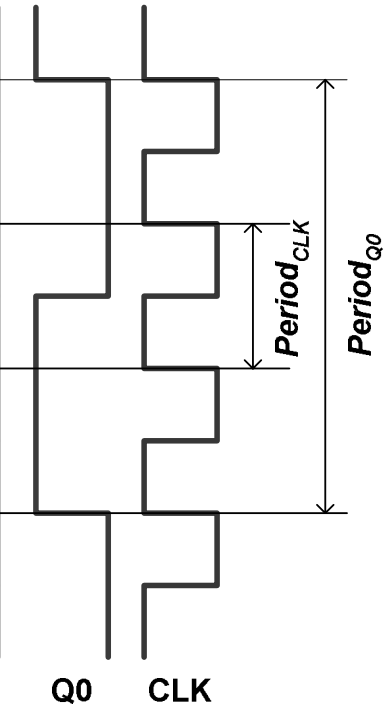




Digital frequency dividers and prescalers. The Johnson Counter (JC) can be designed by insertion the inverter into the feedback of the linear shift register. If JC has the initial zero seed, than period is equal to 2**N*, where ***N*** is the bit-length of the counter.



Initial seed				
0	0	0	0	
1	0	0	1	
2	0	1	1	
3	1	1	1	
4	1	1	0	
5	1	0	0	
0	0	0	0	
1	0	0	1	
2	0	1	1	



The maximal stable period of 2**N* has only JC with the bit-length of ***N***, where ***N***=2****i***, ***i***=1,2,3… . In other cases period depends on the initial seed.

Period$_{JC}$=2

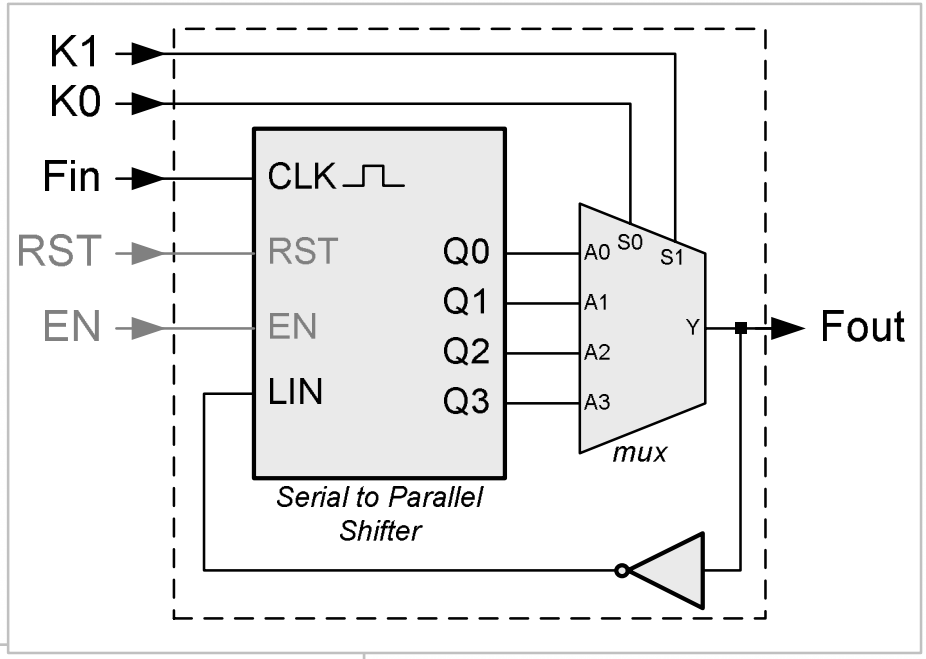
Initial seed				
0	1	0	1	
1	0	1	0	
0	1	0	1	

Period$_{JC}$=8

Initial seed					
0	1	0	1	0	
1	0	1	0	0	
2	1	0	0	1	
3	0	0	1	0	
4	0	1	0	1	
5	1	0	1	1	
6	0	1	1	0	
7	1	1	0	1	
0	1	0	1	0	

SEED(Period$_{JC}$/2)= ——— =INITIAL_SEED

Digital frequency dividers and prescalers. The configurable frequency divider on the base of JC may has next schematic:



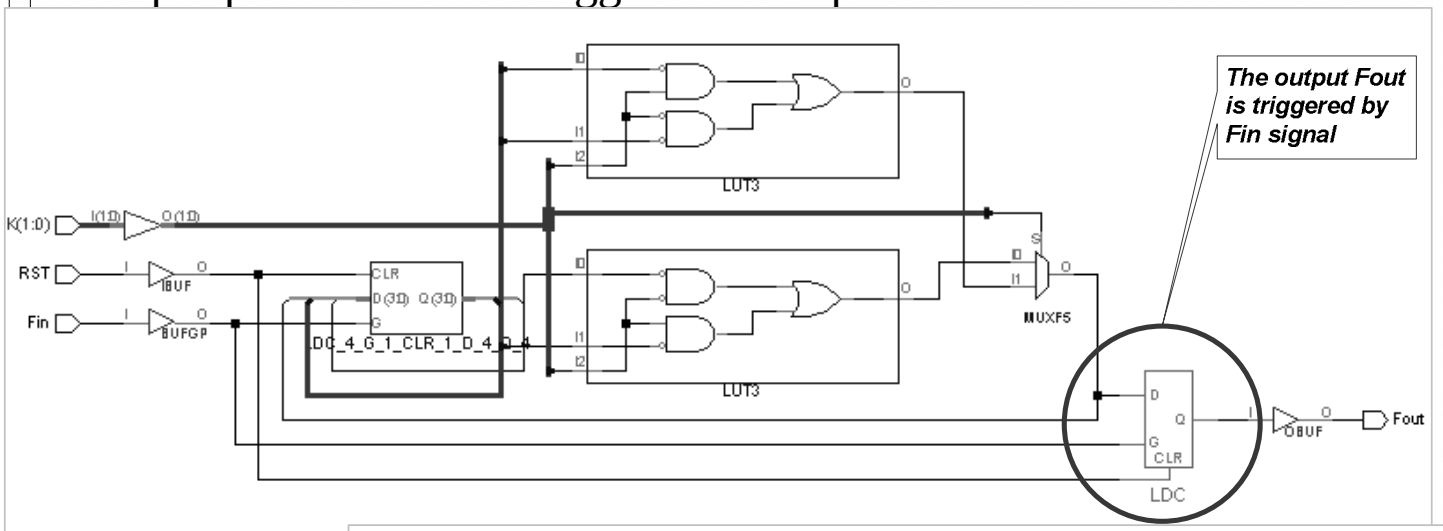

```
--
ARCHITECTURE beh OF jc4 IS
   signal Q: std_logic_vector(3 downto 0);
   signal feedback: std_logic;
BEGIN
   JCOUNTER: process(Fin,RST,Q,feedback)
   begin
      if RST='1' then
         Q<=(others=>'0');
      elsif Fin'event then
         Q(3 downto 1)<=Q(2 downto 0);
         Q(0)<=not feedback;
      end if;
   end process;

   feedback<=Q(CONV_INTEGER(K));

   Fout<=Q(0);

END beh;
```

Possible, but not good

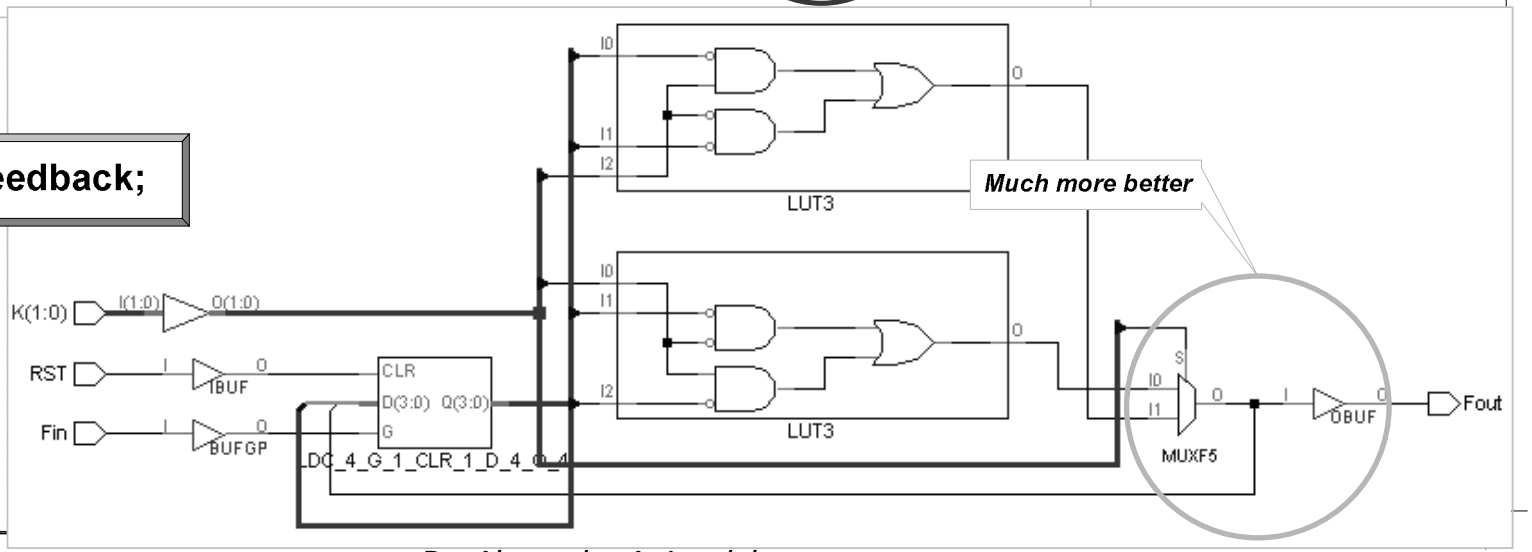
K1	K0	K_{ratio}
0	0	1
0	1	2
1	0	3
1	1	4

```
ENTITY jc4 IS
   port(
      Fin,RST: in std_logic;
      K: in std_logic_vector(1 downto 0);
      Fout: out std_logic
   );
END jc4;
```

Digital frequency dividers and prescalers. In the previous VHDL description the output port **Fout** has a triggered description.



Fout<=not feedback;



Digital frequency dividers and prescalers. But problems are coming! There is an error message during the implementation of the JC VHDL description for the SPARTAN-3 FPGA from Xilinx:

ERROR: Xst: 797: Unsupported Clock statement

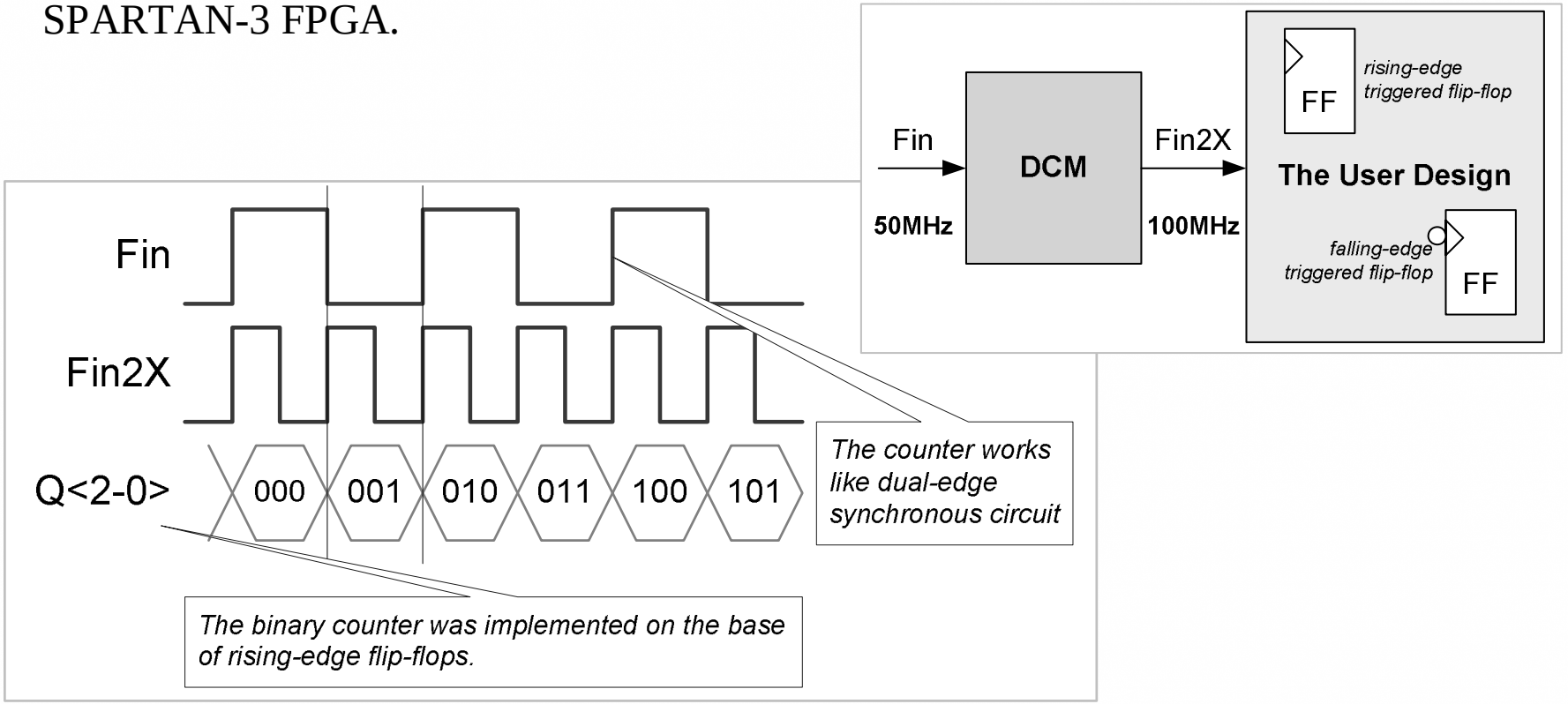
```
...
elsif Fin'event then
...
```

SPARTAN-3 FPGA does not supports the dual-edge triggered flip-flops descriptions (only CoolRunner-II FPGA has dual-edge flip-flops).

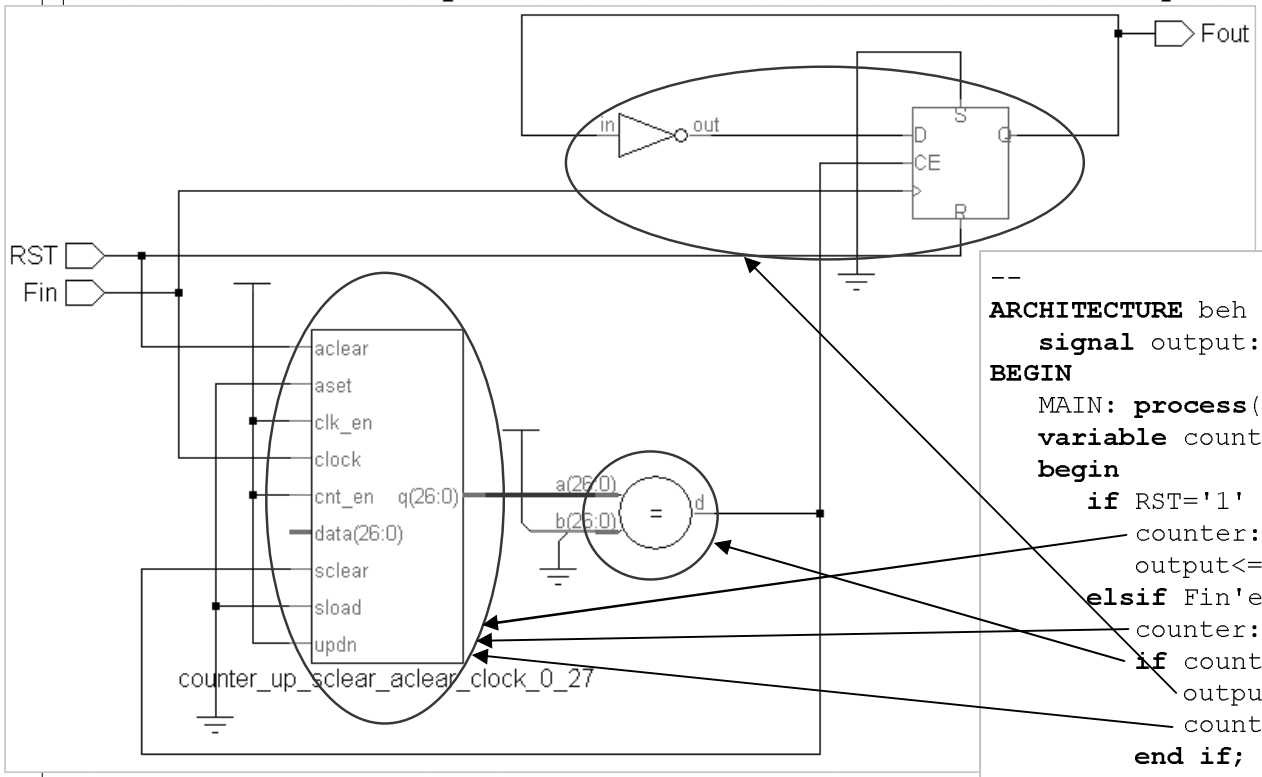
```
JCOUNTER: process(Fin,RST,Q,feedback)
variable Fin_old: std_logic:='0';
begin
   if RST='1' then
      Q<=(others=>'0');
   elsif Fin_old/=Fin then
      Fin_old:=Fin;
      Q(3 downto 1)<=Q(2 downto 0);
      Q(0)<=not feedback;
   end if;
end process;
```

One of the possible solution is to describe the triggered circuit for the incoming clock signal **Fin**, which controls the change of the signal (both rising and falling edges).
But the best solution is to use **DCM** module of SPARTAN-3 FPGA. DCM is a **D**igital **C**lock **M**anager that provides flexible and complete control over the clock frequency in FPGA.

Digital frequency dividers and prescalers. DCM provides incoming clock signals phase shifting as well as clock frequency division and multiplication. At the first practical work we shall consider only the usage of DCM as the circuit which can multiply the incoming frequency by 2. The usage of the system clock with the doubled frequency allows the right implementation of rising-edge triggered flip-flops for SPARTAN-3 FPGA.



Digital frequency dividers and prescalers. The last type of frequency dividers is the case of $K_{ratio}=2*i$, where i – is a large integer. For example, input clock signal has a frequency of 100MHz. The outputs must switch with the frequency of 1Hz. The binary counter and comparator can be a solution for such frequency divider.




```
ENTITY Prescaler IS
   port(
      Fin,RST: in std_logic;
      Fout: out std_logic
   );
END Prescaler;
```

```
--
ARCHITECTURE beh OF Prescaler IS
   signal output:std_logic;
BEGIN
   MAIN: process(Fin,RST,output)
   variable counter:std_logic_vector(25 downto 0);
   begin
      if RST='1' then
         counter:=(others=>'0');
         output<='0';
      elsif Fin'event and Fin='1' then
         counter:=counter+1;
         if counter="10111110101111000010000000" then
            output<=not output;
            counter:=(others=>'0');
         end if;
      end if;
   end process;
   Fout<=output;
END beh;
```

INTRODUCTION

- [x] Digital frequency dividers and prescalers
- [ ] Special shift registers
- [ ] Glitches and hazards in digital circuits
- [ ] Useful materials and links

1. Calculate the period of Intel Pentium 4 processor system clock (p.8)

2. Write down the VHDL description of 4-bit binary “up” counter which has synchronous reset and synchronous enable.

3. Which K_{ratio} it is possible to achieve if the binary counter (p.14) could have the dual-edge synchronization?

4. Which period has 9-bit Johnson Counter with the initial seed of 111111111?

5. If Johnson Counter could be rising-edge triggered (p.17) which shapes of the output signals will be possible to generate?

6. Where is the inverter on both schematics (p.19)?

7. Try to write your own dual-edge triggered D flip-flop VHDL description. Simulate it, synthesize it, and implement it for the SPARTAN-3 FPGA.