

An improved Line-Drawing Algorithm for Arbitrary Fractional Frequency Divider/ Multiplier Based on FPGA

Jian-guang SHEN¹, Tao TAO^{1,2,*}, Xue-song MEI^{1,2}, Mu-xun XU¹ and Shan-hui LIU¹

¹Dept. of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049- China

²State Key Laboratory for Manufacturing Systems Engineering., Xi'an Jiaotong University, Xi'an 710049- China

Received 6 July 2013; Accepted 23 October 2013

Abstract

The proposed algorithm aims to realize the division of arbitrary fractional frequency. The general frequency divider is comprised of the counter cascade, and can realize the even frequency division easily, but it is difficult to realize odd frequency divider and fractional frequency divider. This paper proposes a design method of Arbitrary Fractional Frequency Divider (AFFD) based on FPGA circuit cores by an improved Bresenham line-drawing algorithm, which reduces the hardware resource and eases the implementation process. Bresenham line-drawing algorithm is introduced briefly in this paper outside its original applications. The circuit is designed with Verilog Hard Design Language (VHDL), and the program is proposed. The frequency divider is simulated in Quartus II and the divider is examined in FPGA-Based gear hobbling machine CNC system to control the step-servo motor, and the result is steady and reliable. This paper extends the investigation on the frequency division by the improved algorithm. The algorithm designs arbitrary Fractional Frequency Multiplier, and it is also examined by simulation and experiment.

Keywords: FPGA, Verilog, Fractional fractional frequency divider, Bresenham algorithm, Frequency multiplier

1. Introduction

Frequency divider is one of the most key circuit in the design of digital logic circuit, which is used to divide the given frequency to gain the required frequency [1]. The key technology for the design of a frequency divider is to find a function between the input and output. The general frequency divider is comprised of the counter cascade, and can realize the even frequency division easily [2-5], but it is difficult to realize odd frequency divider and fractional frequency division. Tian hongli [6] has proposed a method of Controllable Arbitrary Integer Frequency Divider which uses the Verilog HDL source code to synthesize a FPGA. But this method only uses in integer frequency divider.

The Bresenham algorithm is an ubiquitous algorithm in computer graphics as it provides a fast incremental interpolation scheme originally used for line plotting. Its major advantages are the elimination of expensive multiplications and divisions as well as the numerical scaling to integer-only arithmetic. The first implementation of this algorithm is due to Jack E. Bresenham and was written for an IBM1401 controlling a plotter [7]. Thomas B. Preußner and Steffen K [8] has proposed a method using the Bresenham algorithm, but which could not be used in any given fractions.

AFFD is realized by a new algorithm which is improved from bresenham algorithm based on Verilog, which has

many appropriate functional for describing the behavior of electronic components ranging from simple logic gates to complete microprocessors and custom chips. In section 2, the basic principle of the bresenham algorithm and the improved algorithm are described. Circuit simulation and experiment are presented in section 3. Then the theoretical errors are discussed in the section 4. Also experimental results are compared with the simulation studies and analytical results in the same section. In section 5, the frequency multiplier is tried to be designed under the proposed algorithm, and it is simulated in Quartus II with Verilog-HDL. Finally, conclusions are drawn in section 6.

2. Basic principle of frequency divider

A fractional frequency divider is an electronic circuit which takes an input signal with a frequency f_{in} and generates an output signal f_{out} [9]:

$$f_{out} = \frac{Num}{Den} f_{in}$$

where Num and Den are integers greater than 0, Num is less than Den.

The Bresenham line algorithm is an algorithm which determines which order to form a close approximation to a straight line between two given points (x_0, y_0) and (x_1, y_1) . It is commonly used to draw lines on a computer screen, as it uses only integer addition, subtraction and bit shifting, all of

* E-mail address: shenjianguang.com@163.com

which are very cheap operations in standard computer architectures. It can be described as follows:

```
function line( $X_0, X_1, Y_0, Y_1$ )
int  $\Delta x := X_1 - X_0$ 
int  $\Delta y := Y_1 - Y_0$ 
real error := 0
real  $\Delta err := \text{abs}(\Delta y / \Delta x)$  // assume  $\Delta x \neq 0$ 
int  $y := Y_0$ 
for x from  $X_0$  to  $X_1$ 
plot(x,y)
error := error +  $\Delta err$ 
if (error  $\geq$  0.5) then
 $y := y + 1$ 
error := error - 1.0
```

The slope $\Delta y / \Delta x$ depends on the endpoint coordinates only and can be precomputed, and the ideal y for successive integer values of x can be computed starting from y_0 and repeatedly adding the slope. In practice, the algorithm can track, instead of possibly large y values, a small error value between -0.5 and 0.5 : the vertical distance between the rounded and the exact y values for the current x . Each time x is increased, the error is increased by the slope; if it exceeds 0.5 , the rasterization y is increased by 1 (the line continues on the next lower row of the raster) and the error is decremented by 1.0 .

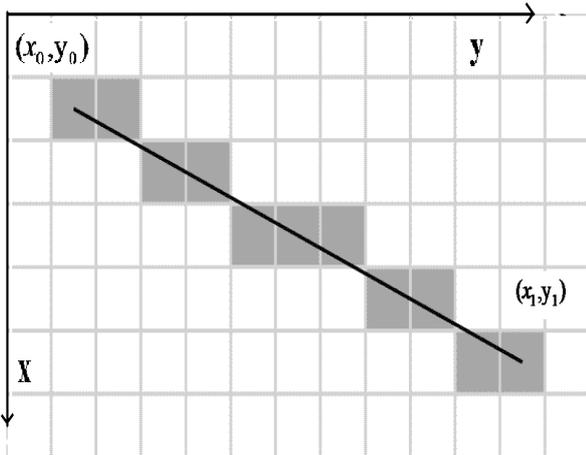


Fig. 1. Illustration of the result of Bresenham's line algorithm.

So, if we build up a line from $(0, 0)$ to $(11, 5)$ by the Bresenham line-drawing algorithm 11 pixel points should be plotted that 11 units along the y axis and 5 units along the x axis.

$$y = \frac{y_1 - y_0}{x_1 - x_0} = \frac{Num}{Den} \cdot x = \frac{5}{11}x$$

Likewise, based on this algorithm we can see y as output pulse, x as input pulse that the frequency 11MHz can be rationally divided into 5MHz.

Although x and y are integers, floating-point computation is necessary, which is a waste of hardware resources and reduces the computing speed. In order to solve this problem, a new algorithm is presented as follows.

```
function line( $X_0, X_1, Y_0, Y_1$ )
int  $\Delta x := X_1 - X_0$ 
int  $\Delta y := Y_1 - Y_0$ 
real error := 0
real  $\Delta err := \Delta x$  // assume  $\Delta x \neq 0$ 
int  $y := Y_0$ 
for x from  $X_0$  to  $X_1$ 
plot(x,y)
error := error +  $\Delta x$ 
if (error  $\geq$   $\Delta y$ ) then
 $y := y + 1$ 
error := error -  $\Delta y$ 
```

3. Frequency division

Based on the improved Bresenham algorithm, the frequency divider is realized in FPGA with the Verilog HDL, which is depicted in fig1. Let Num and Den have no common factors, the initialization condition of inherited from the line drawing algorithm for the Err register can be relaxed. The addition of Num and conditional subtraction of Den is easily identified as addition modulo Den. When input clock pulse is coming the Num is added to the Err register. When the Err is bigger than the Den, the system output a pulse and then Den is subtracted from Err. Through such a cycle, the input clock pulse f_{in} is successfully divided into f_{out} , which is prevented and illustrated for a 5/13 scaling in fig 5 and fig 6.

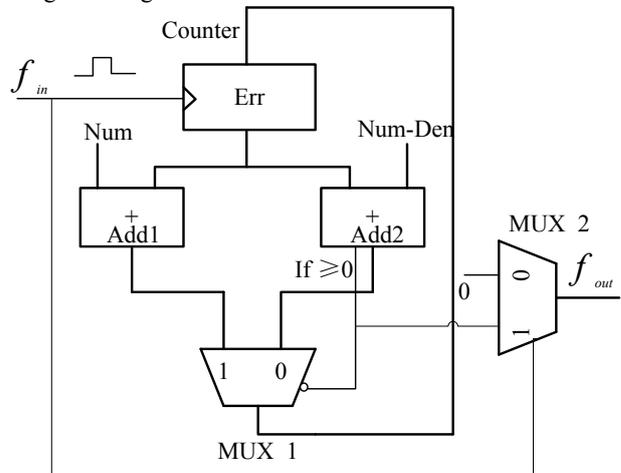
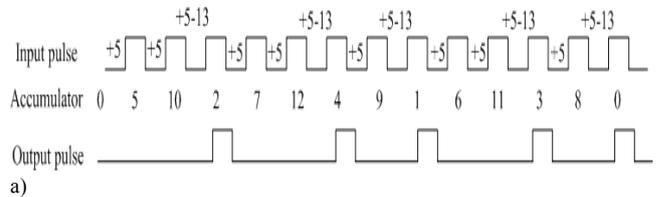
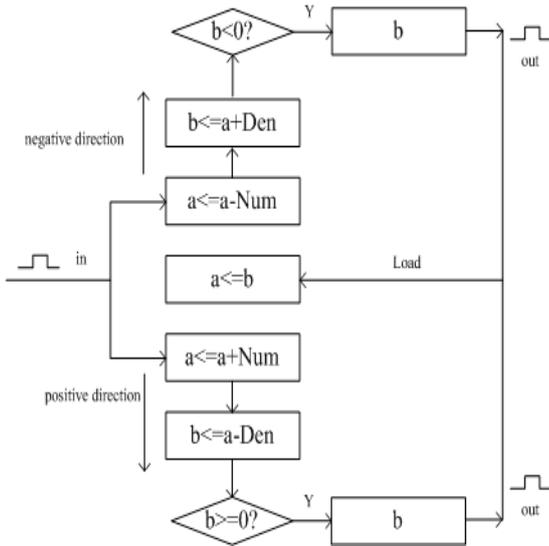


Fig. 2. Improved Bresenham Frequency Divider

According to the method, the design flow of FPGA is shown as follow in Fig 3.





b)
Fig. 3. Design flow of FPGA of frequency division

When a pulse comes, in order to make sure the accumulator a does not produce retention pulse error, the different accumulators share a same register. When the pulse direction is positive that the accumulator adds Num. And when the accumulated value is over Den, then the divider outputs one positive pulse besides the value of accumulator reduces a Den. On the contrary, when the pulse direction is negative that the accumulator reduce a Num. And when the accumulated value is less than 0, the divider output one negative pulse besides the value of accumulator adds a Den. Input pulse can be divided into output pulse by analogy.

The simulation of the frequency divider is simulated in the next part.

From the previous discussion in Fig.2, we can found that easily the initialization value of Err is not critical. Because the output is a single pulse, that there is a necessary error between the input and the output. Further the maximum error is 1. In order to calculate the error of output, the input pulse is counted from 0. Assume: when the input pulse is f_{in} , the output pulse is 0. And when the input pulse is $f_{in} + 1$, the output pulse is 1. Therefore, the f_{in} is the maximum input error corresponding to the maximum output error.

$$\begin{cases} f_{in} \times Num < Den \\ (f_{in} + 1) \times Num \geq Den \\ Num, Den \in Z \end{cases}$$

we can get:

$$f_{in} = \left\lfloor \frac{Den-1}{Num} \right\rfloor$$

The frequency error is calculated as:

$$\Delta Error = f_{in} \times \frac{Num}{Den} = \left\lfloor \frac{Den-1}{Num} \right\rfloor \cdot \frac{Num}{Den}$$

From the discussion we can found that the maximum error of output pulse of the divider is always less than 1, and it changes with the input pulse cyclely.

4. Circuit Simulation

The AFFD circuit is realized and tested on Altera FPGA development board with EP1C12Q24076N device. It is simulated on Quartus II, and the simulation circuit synthesized as fig 4

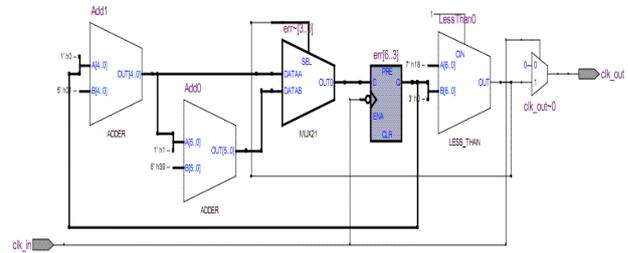


Fig. 4. Circuit of the AFFD

The whole device consumes only 1% of available logic cells. In order to identify that the divider can work correctly, we do the experiment about some fractions. The simulation waveforms are shown in Fig 4.

Given the fraction division coefficient is $\frac{5}{13}$, and the input pulse clock frequency is 100Mhz. The simulation result is as follow shown in fig.5:

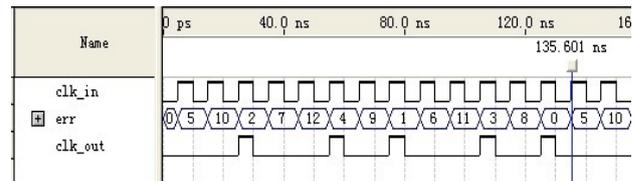


Fig. 5. Simulation result of frequency division

Experiment on the gear hobbling machine CNC system which is designed based on FPGA and MCU (Fig 6) verified well with the simulation result which is shown in Fig 7. The speed of hob spindle and work piece spindle is controlled by this system. And the control instructions of the speed of Hob spindle is given as 60 mm/min. The result in fig.8 shows that ratio of the speed of hob spindle to workpiece spindle equals the fraction division coefficient.

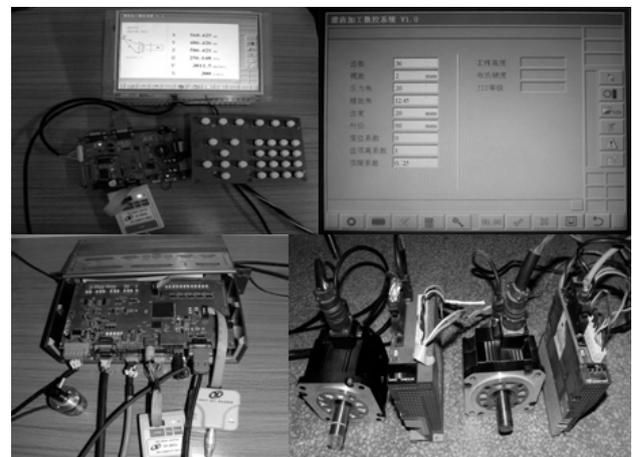


Fig. 6. Gear hobbling machine CNC system

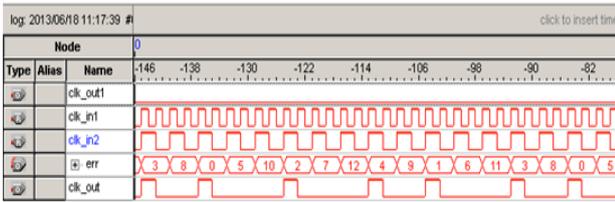


Fig. 7. Frequency division experiment result

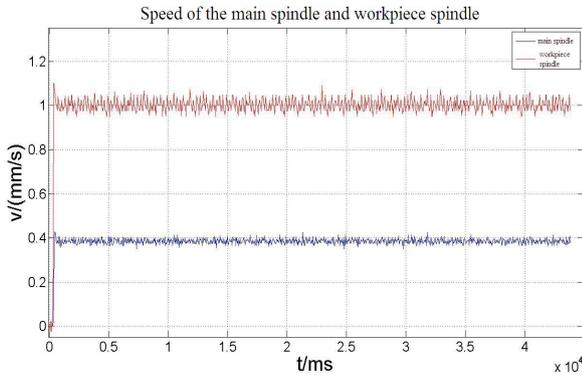


Fig. 8. Speed control experiment result

5. Design of Fractional Frequency Multiplier

A frequency multiplier is generally a device for multiplying by an integer the frequency of a circuit. It is an electronic circuit that generates an output signal of which the output frequency is a multiple (or harmonic) of the input frequency.

$$f_{out} = \frac{Num}{Den} f_{in}$$

where Num and Den are integers greater than 0, Num is bigger than Den.

Frequency multipliers consist of a nonlinear circuit which distorts the input signal and generates multiples of the input signal. A bandpass filter selects the desired harmonic frequency and removes the unwanted fundamental as well as other harmonics from the output.

In order to get the transmission ratio of the Frequency Multiplier greater than 1, that must achieve the frequency doubled function which is for the any fractional of the input pulse signal. Currently, with the technical of frequency multiplication is more complicated than the technical of frequency division, the phase-locked loop technology can justly, output the frequency doubled pulse which meet the requirements of duty cycle and phase; but for the uncertain input pulse frequency, the phase-locked loop technology can not get the precise real-time double frequency.

Because of the varying frequency input pulse, the system can not predict the time of next arrival pulse, and there for the system can not output uniform frequency pulse in time. To get more uniform output pulse, the system should not output until the next pulse comes. Therefore there is a delay of one input pulse resulting the output delay $\frac{Num}{Den}$ bigger than one pulse. And the real-time becomes relatively poor.

To output uniform frequency pulse in a certain period, a frequency division coefficient is assigned as $\frac{Num}{N \cdot Den}$,

which gets form the $\frac{Num}{Den}$ dividing by the cumulative

number N of the basic frequency f'_0 of this certain period.

The coefficient is used to divide the basic frequency. Then the frequency multiplication is turned into frequency division. To meet the requirements of the frequency division, the basic frequency should be smaller than the max frequency of the system. Because of the divider is used that it is more difficult to be realize by the hardware. The principle is shown in Figure 9.

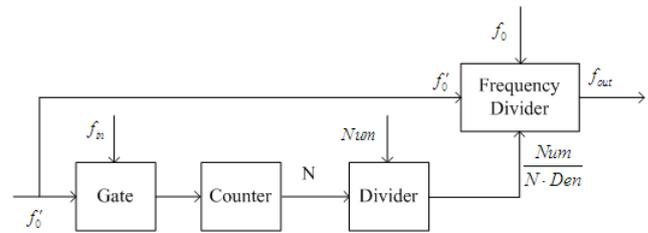


Fig. 9. Schematic diagram using frequency divider

There is another way to avoid the usage of divider that two frequency divider is instead of. It needs two steps: firstly realize the frequency division with the Division factor

$\frac{Num}{N}$ and secondly realize the frequency division with the

division factor $\frac{1}{Den}$. The principle is shown in Figure 10.

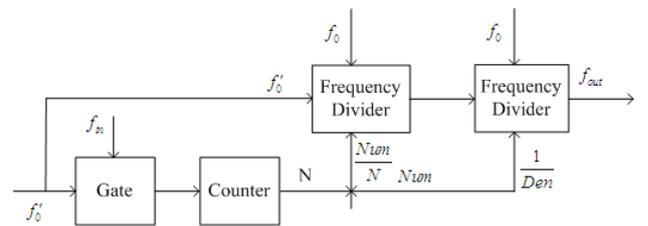
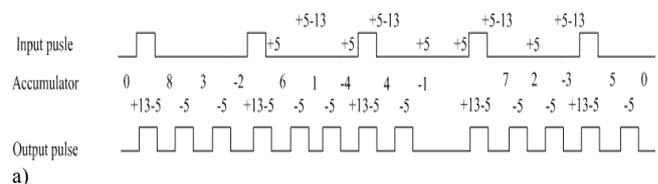


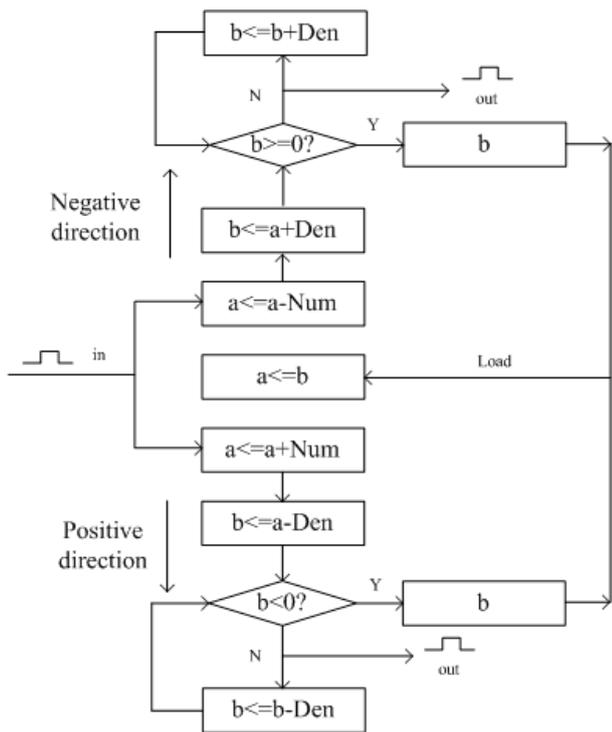
Fig. 10. Schematic diagram using two frequency divider

Besides this method also has limitations: in order to ensure dividing frequency, the fundamental frequency must be high enough to ensure that the pulse cycle count value N is greater than Num , which limits the range of the multiplier.

Because the multiplier in the realization of electronic gear has a lot of limitations and drawbacks, so the experiment equipment of the gear hobbing CNC system mentioned in last part is not designed in motion control. It is used in the hand veins control which does not require too high uniform pulse. And the system continues to use the method, which is similar to AFFD to realize the frequency multiplication.

The design flow of FPGA is shown as follow in Fig 11.





b)
Fig. 11 Design flow of FPGA of frequency multiplication

The simulation of the frequency multiplier is shown as in Fig.12.

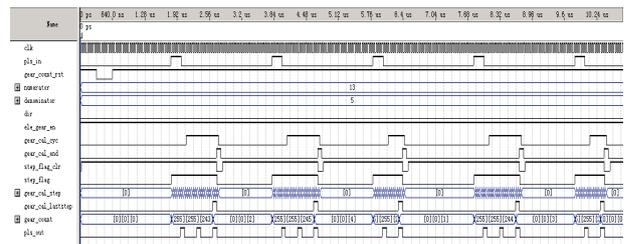


Fig. 12 simulation result of frequency multiplication

6. Conclusions

In this paper, an improved Bresenham Line-Drawing Algorithm has been presented and applied to fractional frequency division in hardware description language. The frequency divider is synthesized and simulated with Quartus II. According to the simulation result based on FPGA, it is concluded that the divider can divide the input pulses into target output pulses. Because of its integer-only arithmetic algorithm, it eliminates expensive multiplication and divisions.

Furthermore, it is advantageous in terms of logic cell consumption and achievable reference impulse frequency. In addition, simulation result is consistent with the experimental result. Besides it also provides a way for frequency multiplication, which is proved by simulation and experiment.

References

- Hou Li-gang, Zhang Tian-ran, "A High-Accuracy Design of Frequency Divider With FPGA And ASIC Implementation", Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on, 23-25 Aug. 2012 , pp.1653-1656
- Brennan P.V., Jiang, D, Wang, H, "Memory-controlled frequency divider for fractional-N synthesizers", Electronics Letters, 12th October 2006, Volume: 42, Issue: 21, pp.1202-1203
- Kim, K.Y, Min, Y.J., Kim, S.W, Park, J., "Low-power programmable divider with a shared counter for frequency synthesizer". IET Circuits Devices Syst. 2011, Volume: 5 ,Issue:3, pp. 170-176
- Milan, Stork. "Digital Fractional Frequency Synthesizer Based on Counters", Turk J Elec Engin, 2006, Volume:14, Issue: 3, pp. 387-397
- Milan Stork. "Counter Based Frequency Synthesizer", XVII Imeko World Congress Metrology in the 3rd Millennium, June 2003, pp. 22-27
- Hongli, T; Shuo, S. "Controllable Arbitrary Integer Frequency Divider Based on VHDL". 2009 International Joint Conference on Artificial Intelligence, 2009, pp.691-694
- Jack E. Bresenham. "Algorithm for computer control of a digital plotter". IBM Systems Journal, 1965, Volume: 4, Issue:1, pp.25-30
- Preuber TB, Spallek RG. "Analysis of a Fully-Scalable Digital Fractional Clock Divider". International Conference on Application-specific Systems, Architectures and Processors Sep, 2006, pp. 173-177
- Liangling, GU; Nanquan, ZHOU, "Research of Frequency Divider Based on Programmable Logic Device", Procedia Environmental Sciences 10 , 2011 , pp.820 - 824
- Ward, J,Jet, G, Maestrini, "Capability of THz sources based on Schottky diode frequency multiplier chains", Microwave Symposium Digest, 2004, Volume:3, pp. 1587-1590.
- E. Kaplani, "Design and Performance considerations in stand-alone PV powered telecommunication systems", Journal of Engineering Science And Technology Review 5(1),2012, pp.1-6.
- Sh. Mirseidova, A. Inoue, L. Atymtayeva, "Evaluaton of Fair Market Price of Resources In Oil And Gas Industry Using Fuzzy Sets And Logics", Computer Modelling and New Technologies.16(1),2012,pp.30-34.