An Intuitive Signal Processing Approach for Temperature Fluctuations in Fuel Subassemblies

Paawan Sharma
RTSD, EIG, IGCAR
Kalpakkam-603102
INDIA

N. Murali
RTSD, EIG, IGCAR
Kalpakkam-603102
INDIA

P Mohanakrishnan
RPD, REG, IGCAR
Kalpakkam-603102
INDIA

P Swaminathan
EIG (Retd.), IGCAR
Kalpakkam-603102
INDIA

ABSTRACT

This paper describes the approach towards signal processing characteristics for temperature fluctuations from a fast thermocouple located above fuel subassemblies of fast reactor. Simulated temperature profile denoting various power levels was fed to the FPGA and the RMS value at each power level was calculated in real time. The technique mentioned in the paper helps in the analysis of reactor power and also fluctuation in it due to subassembly blockage. Altera Cyclone III FPGA was used as target device for Terasic DE0 board.

Keywords

Temperature fluctuations, root mean square, subassembly. Fast reactor.

1. INTRODUCTION

Temperature fluctuations are generated in a subassembly due to power increase or a blockage. These fluctuations are very important for studying the safety aspects. In case of power increase, fluctuations are generated because of improper mixing of comparatively cold and hot fluid flow around the subassembly, whereas in case of blockage, fluctuations are generated due to less coolant flow through subassembly. In both the cases, there is a gradual increase in thermal power. Our main intention here is to correlate the increase in thermal power to its RMS value. From safety point of view, blockage detection is very important [1] as it can lead to more serious faults because of high power density of the core [2]. K-type bare thermocouple is used for measuring temperature [3]. The digitized data is fed as input to FPGA development board.

2. THEORY FOR FLUCTUATION GENERATION IN A SUBASSEMBLY

Figure 1 & 2 shows the phenomenon behind the fluctuation generation under normal condition when power is increased slowly as well as in blockage respectively. The latter is different in the sense that the amplitude level of fluctuations is more, hence leading to a greater RMS value. The presence of Vortex Street [4], which are repeating pattern of swirling vortices, is responsible for fluctuation generation.

So far, many works are reported where the use of temperature fluctuation based parameters are being proposed. One such study by Tsunoda [5] supports the use of fluctuations. Also, Greef [6] calculated the parameter values for different thermodynamic conditions.

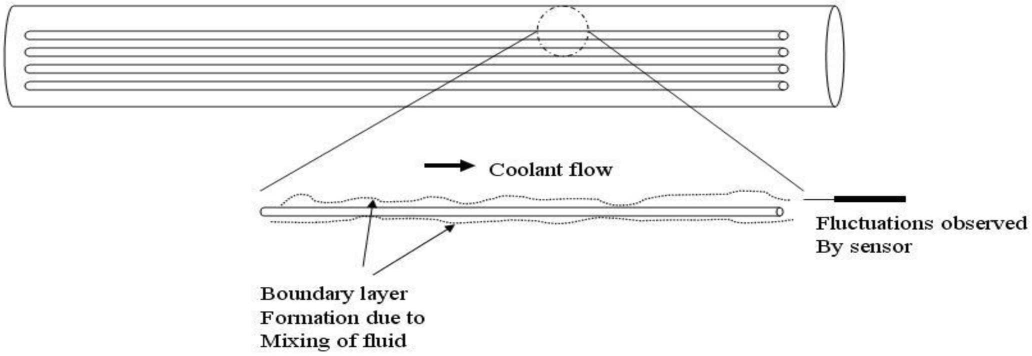

Fig 1: Fluctuations due to thermal power increase

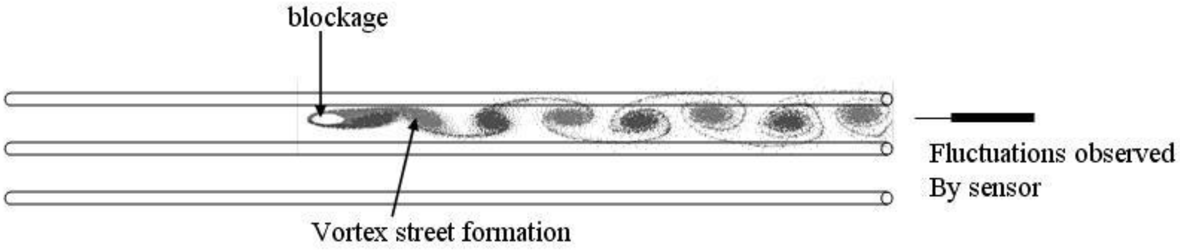

Fig 2: Fluctuations due to blockage

3. SIGNAL PROCESSING SCHEME

Figure 3 shows the schematic for the signal processing of such fluctuations. It consists of a Band pass filter and RMS calculator. The specifications for band pass filter are mentioned in TABLE I.

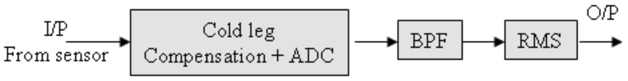

Fig 3: Processing Sequence

A band pass filter is required in order to remove noise pickup during data transmission as well as the noise introduced at the reactor inlet. The data sequence is generated artificially which resembles to that obtained from a fast k-type thermocouple whose response time is 150ms. A higher response time thermocouple is not useful for such a processing as it requires a large amount of time to process the data.

Table 1. BPF Specifications

S. NO.	Band pass filter Specifications	
1	f_L Lower cutoff frequency	0.01Hz
2	f_H Upper cutoff frequency	24 Hz
3	Sampling Frequency F_s	100 Hz

The digital filter design is accomplished by using fir_design script written in SCILAB, which calculates the filter coefficients. The same script can be written for MATLAB also.

4. TOWARDS FPGA IMPLEMENTATION

Implementing signal processing scheme on FPGA showed an increase in overall magnitude performance [7]. Also, its reconfigurable capability is an added advantage. The signal processing scheme shown in fig. 3 was implemented on Altera® Cyclone III family FPGA development board DE0 [8], which is equipped with EP3C16F484C6 FPGA device, comprising of 15,408 LEs(Logic Elements). The board provides 346 user I/O pins. Fig. 4 shows the block diagram for DE0 board.

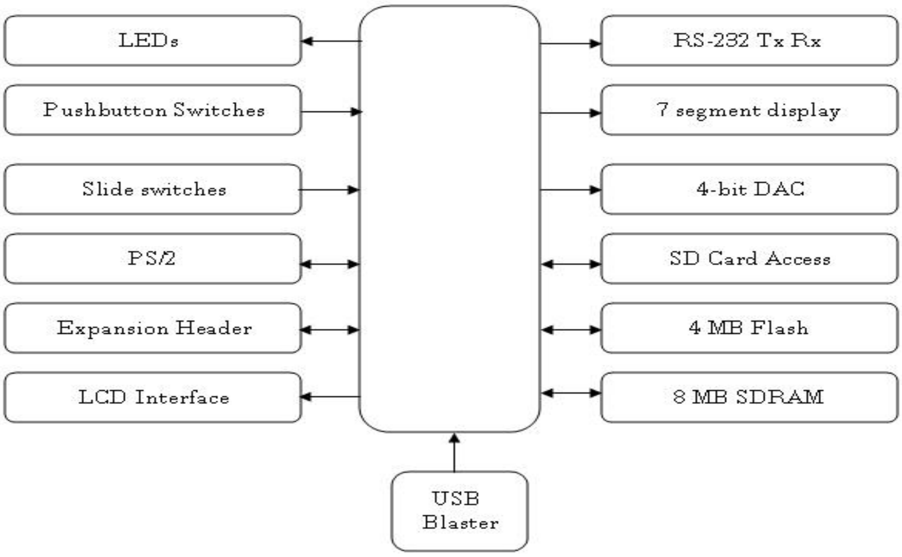

Fig 4: DE0 Block Diagram

4.1 Calculation of filter Coefficients

Equation (1) which represents the frequency response of ideal band pass filter is used for calculating the coefficients for a rectangular window. Alternatively, various MATLAB/SCILAB commands like fir1, fir2 etc. can also be used for coefficient calculation. Fig. 5 shows the frequency response of the filter.

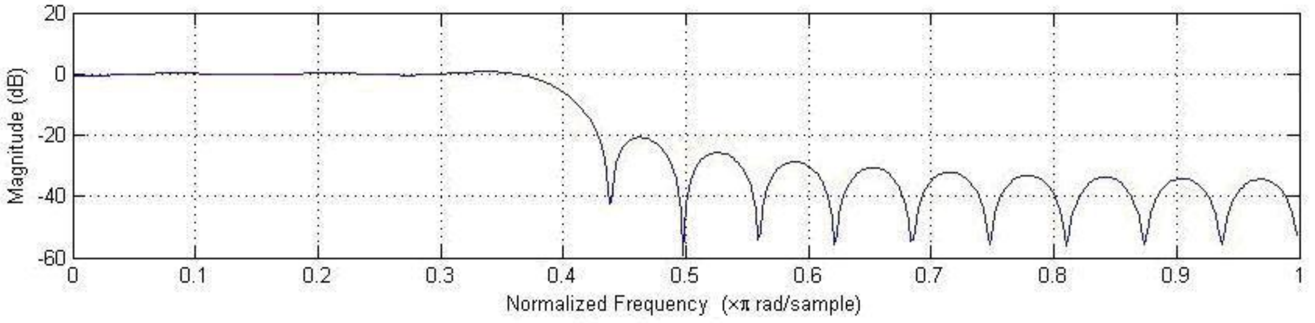

Fig 5: Filter Response

The rectangular window gives the minimum integrated squared error from the ideal response [9]. The root mean square (RMS) block denotes the statistical measure of the magnitude of varying filtered temperature fluctuations, and is mathematically represented as,

$$x_{rms} = \sqrt{\frac{x_1^{\ 2} + x_2^{\ 2} + \dots\dots + x_n^2}{n}}$$

where n denotes the bin size taken for every individual calculation. For obvious reasons, n =2 for calculating running RMS, where the time separation between two successive values of temperature is equal to the sampling period of the ADC block.

$$h_d[n] = \left[\frac{\sin\{\omega_{c2}(n-M)\}}{\pi(n-M)} - \frac{\sin\{\omega_{c1}(n-M)}{\pi(n-M)} \right] \quad ; n \neq M$$

$$= \frac{\omega_{c2} - \omega_{c1}}{\pi} \quad ; \ n = M \qquad (1)$$

where, ω_{c2}, ω_{c1} = cutoff frequencies;

M = filter order/2;

4.2 Calculation of number of precision bits

A script cal_sfix_bits was used to calculate the bit strength needed for specific magnitude and precision. Figure 6 shows the relationship between precision and number of factional bits.

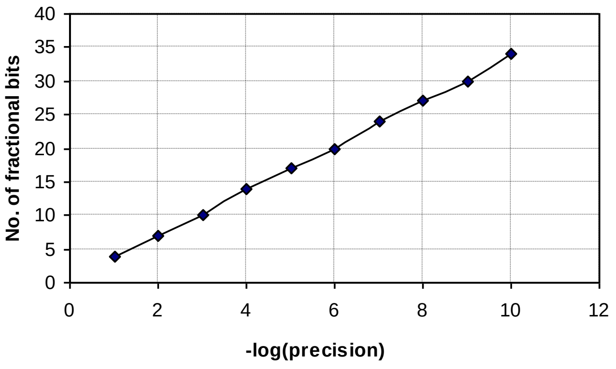

Fig 6: Precision vs No. of fractional bits

Table II represents a comparison of coefficient values for different fractional bit strengths. It may be required to truncate the output resulted from multiplication, so as to preserve the FPGA resources. This is performed by keeping m LSBs of integer and n MSBs of fractional portion from 2m and 2n bits of the output. For the sake of simplicity, however, the bit pattern is used unaltered.

Table 2. Coefficient Conparison

Coefficient value (Calculated)	Actual value	
	7 bit	14 bit
0.048455588950058	0.046875	0.048401
-0.001999671029421	0.0078125	-0.0020142
0.300730704615709	0.29688	0.30072
-0.064365833819427	-0.070313	-0.064392
0.017947456292925	0.015625	0.017944
0.315681787493831	0.3125	0.31567

Coefficient value (Calculated)	Actual value 7 bit	 14 bit
0.478000000000000	0.47656	0.47797
0.106223795893264	0.10156	0.1062
0.021789901874715	0.015625	0.02179

Fig. 7 shows block level model. The input and output are the boundary ports used for giving stimulus and checking the output on a scope. The input is defined by 16 bits {5:11} i.e. 5 integer and 11 fractional bits. Coefficient values are also represented by 16 bits {2:14} Fig. 8 shows the details of the BPF FIR filter subsystem, where the filter is implemented in direct form.

Since any FIR filter can be implemented using direct form, non-recursive structure, it is always possible to implement a FIR filter non-recursively. The main advantage of direct form is that a constant load is observed by each delayed input signal, buffers are not required [10]. Gain blocks represent the corresponding filter coefficient values. Fig. 9 shows RMS implementation.

5. SIMULATION RESULTS

Software simulation results were obtained by feeding noise added sine wave as a input to FPGA. Fig. 10 shows the input and output for sine wave of different frequencies viz. 1,5,15 & 30 Hz added with random noise having mean value as 232 and 0.1 as variance. It is to be noted that in a certain frequency range, the value of processed signal parameter (fluctuations) is proportional to the frequency of the fluctuations.

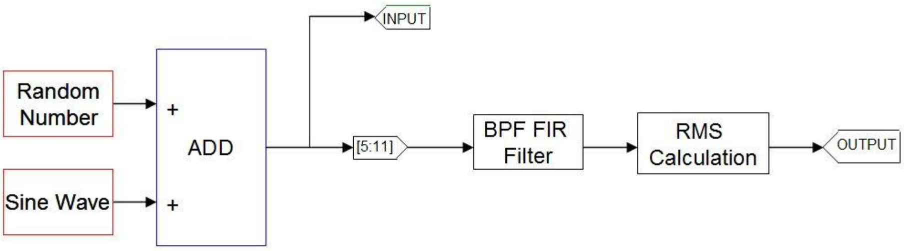

Fig 7: Block level model

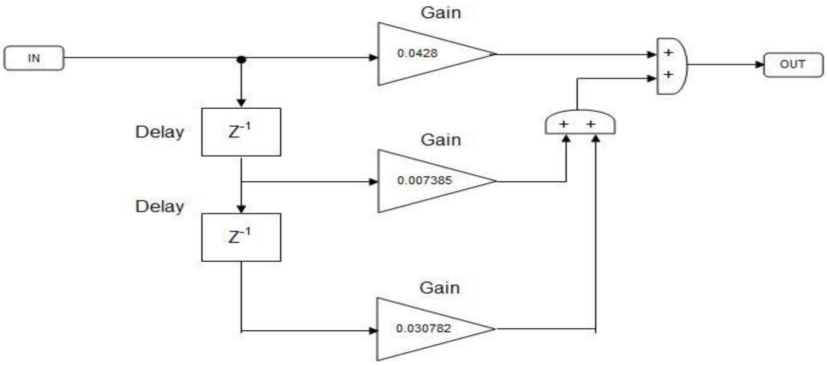

Fig 8: A section of direct form FIR filter

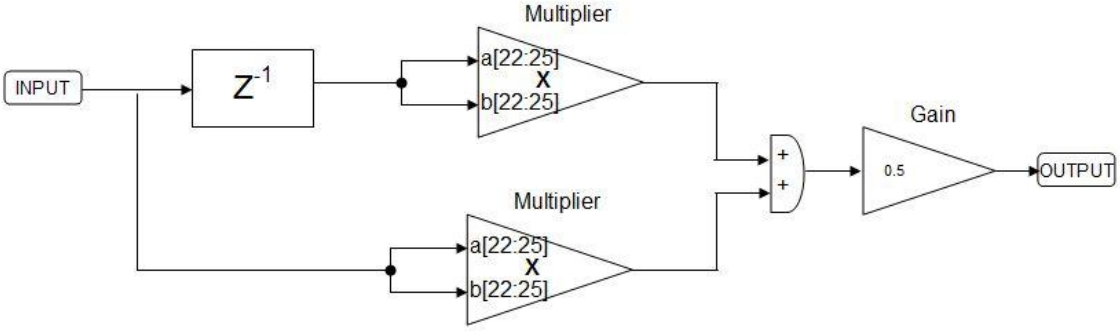

Fig 9: RMS block with bin size=2

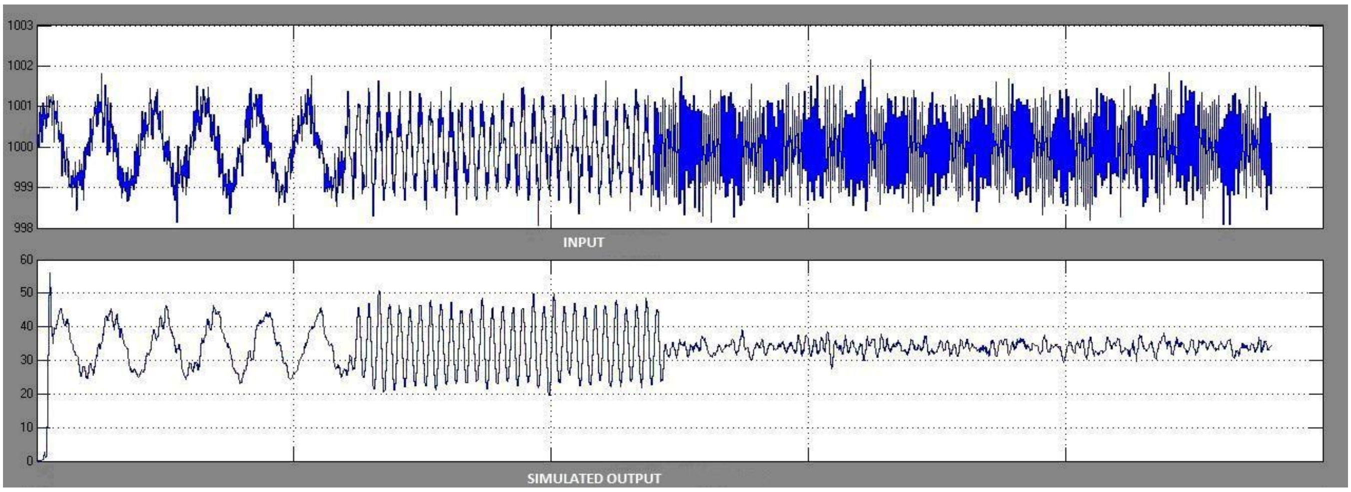

Fig 10: Simulation Results

6. CONCLUSION

A signal processing scheme was modeled and simulated for a specific spectrum of signals added with noise, and the number of fractional bits required for correct number representation was estimated. It is to be noted that a statistical parameter can be derived from the processed signal which would help in recognition of calculative abnormalities in power. A study is proposed in future for further investigation. The scripts can be referred from the Appendix.

7. REFERENCES

[1] Gregory, C.V.; Lord, D.J., “Study of local blockages in fast reactor sub-assemblies,” Journal of Brit. Nucl. Energy Soc., v. 13, no. 3, pp. 251-260, 1974.

[2] Alan Edward Waltar, Albert Barnett Reynolds, “Fast breeder reactors,” Pergamon Press, 1981.

[3] ASTM Committee on Temperature Measurement, “Manual on the Use of thermocouples,” 4th Edition, pp.46, 1993.

[4] Stavros Tavoularis, “Rod bundle vortex networks, gap vortex streets and gap instability: A nomenclature and some comments on available methodologies,” Nuclear Engineering and Design 241, pp. 4612–4614, 2011.

[5] Tomio Tsunoda, “Application of temperature fluctuation to Anomaly detection in Fuel Assemblies", Journal of Nuclear Science and Technology 13(3), pp. 103-110, 1976.

[6] C.P. Greef, “Temperature fluctuations: an assesment of their use in the detection of fast reactor coolant blockages,” Nuclear Engineering & Design 52, pp. 35-55, 1979.

[7] Gabriele D'Antona, Alessandro Ferrero, “Digital Signal Processing for Measurement Systems: Theory and Applications,” Springer Publication, pp.58, 2006.

[8] http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=364

[9] Charles Chein, “Digital Radio Systems on a chip: a systems approach,” Kluwer Academic Publications, pp. 372, 2001.

[10] Russell J. Petersen, L. Hutchings, “An assessment of the suitability of FPGA-based systems for use in digital signal processing”, Lecture Notes in Computer Science, Volume 975/1995, 293-302. 1995.

```
*********************************************
APPENDIX- A : FIR FILTER DESIGN
*********************************************
function [coeff]=fir_design(N, ftype, fc, fs)

// Output variables initialisation (not found in input
variables)
coeff=[];
// Display mode
mode(0);
// Display warning for floating point exception
ieee(1);
// Returns filter coefficients for a rectangular window
function
// N = order of filter
// ftype = "bp", "lp, "hp", "bs"
// for "bp" & "bs" fc = [fc1 fc2]
// fc and fs - cutoff and sampling frequency in Hz
// Example - a = fir_design(12,"bp",2000,10000)
//
M = mtlb_a(round(mtlb_double(N)/2),1);
ft = ftype;
// Calculation of normalized frequency
select ft
  case "lp" then
    fn = ((2*%pi)*mtlb_double(fc))/mtlb_double(fs);
    clear("n");
    for n = mtlb_imp(1,1,mtlb_double(N))
      if mtlb_logic(n,"~=",M) then
        h(1,n) =
matrix(sin(fn*mtlb_s(n,M))/(%pi*mtlb_s(n,M)),1,-
1);
      elseif mtlb_logic(n,"==",M) then
        h(1,n) = matrix(fn/%pi,1,-1);
      end;
    end;
    coeff = h;
    return;

  case "hp" then
    fn = ((2*%pi)*mtlb_double(fc))/mtlb_double(fs);
    clear("n");
    for n = mtlb_imp(1,1,mtlb_double(N))
      if mtlb_logic(n,"~=",M) then
        h(1,n) = matrix(mtlb_s(1,fn/%pi),1,-1);
      elseif mtlb_logic(n,"==",M) then
        h(1,n) = matrix(-
sin(fn*mtlb_s(n,M))/(%pi*mtlb_s(n,M)),1,-1);
      end;
    end;
    coeff = h;
    return;
  case "bp" then
    fn2 =
((2*%pi)*mtlb_double(mtlb_e(fc,2)))/mtlb_double(f
s);
    fn1 =
((2*%pi)*mtlb_double(mtlb_e(fc,1)))/mtlb_double(f
s);
    for n = mtlb_imp(1,1,mtlb_double(N))
      if mtlb_logic(n,"~=",M) then
        h(1,n) =
matrix(mtlb_s(sin(fn2*mtlb_s(n,M))/(%pi*mtlb_s(n,
M)),sin(fn1*mtlb_s(n,M))/(%pi*mtlb_s(n,M))),1,-1);
      elseif mtlb_logic(n,"==",M) then
        h(1,n) = matrix(mtlb_s(fn2,fn1)/%pi,1,-1);
      end;
    end;
    coeff = h;
    return;
  case "bs" then
    fn2 =
((2*%pi)*mtlb_double(mtlb_e(fc,2)))/mtlb_double(f
s);
    fn1 =
((2*%pi)*mtlb_double(mtlb_e(fc,1)))/mtlb_double(f
s);
    for n = mtlb_imp(1,1,mtlb_double(N))
      if mtlb_logic(n,"~=",M) then
        h(1,n) =
matrix(mtlb_s(sin(fn1*mtlb_s(n,M))/(%pi*mtlb_s(n,
M)),sin(fn2*mtlb_s(n,M))/(%pi*mtlb_s(n,M))),1,-1);
      elseif mtlb_logic(n,"==",M) then
        h(1,n) =
matrix(mtlb_s(1,mtlb_s(fn2,fn1)/%pi),1,-1);
      end;
    end;
    coeff = h;
    return;

end;
endfunction
```

```
*************************************************
*
APPENDIX- B : CALCULATION OF FRACTIONAL BITS
FOR FIXED POINT REPRESENTATION
*************************************************
*
function [m, n]=cal_sfix_bits(minimum, maximum,
res, signed)

// Output variables initialisation (not found in input
variables)
m=[];
n=[];

// Display mode
mode(0);

// Display warning for floating point exception
ieee(1);

// calculates the no. of integer and fractional bits
// for a required range = (minimum, maximum)
// and resolution
// Also supports signed/unsigned numbers
// NOTE : For calculating Real World Values
(RWVs)
//       multiply the fixed point equivalent by 2^(-n)
// Example : [c d] = cal_sfix_bits(-1.222,
234.542,.0001,1)
s = [minimum,maximum];
t = abs(mtlb_double(s));
%v0 = t;u = max(%v0,firstnonsingleton(%v0));
if mtlb_logic(mtlb_double(signed),"==",0) then
  a =
floor(mtlb_a(log2(mtlb_double(maximum)),1));
  b = ceil(log2(1/mtlb_double(res)));
else
  a = mtlb_a(floor(log2(mtlb_a(u,2))),1);
  b = ceil(log2(1/mtlb_double(res)));
end;
m = a;
n = b;
endfunction
```

```
***********************************************
*
```