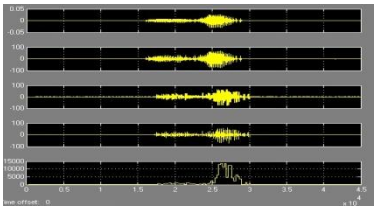
Journal of Computational Science & Engineering 16 (2015) 546-550

Available at http://www.asocse.org ISSN 1710-4068

Speech Endpiont Detection Using DSP Builder

Lihua Zhao

College of Electronic Science, Northeast Petroleum University, Daqing, China



Abstract: **In this paper, we use DSP Builder to realize the speech signal endpoint detection based on short-time energy variation. In the design, we use the LUT (look-up table) window design and FSM (finite state machine) to simplify the hardware circuit and improve the operating speed. Using DSP Builder to preprocess the speech signal (pre-emphasis, framing and windowing),to calculate the signal short-time energy of per frame, to convert the energy to state value according to the result of speech energy of each frame compare with the energy threshold, then determined the endpoint of speech signal by energy threshold and energy logic state numeric sequence. The results of simulation show that the method can detect the beginning point and ending point of speech signal effectively.**

Keywords: Endpiont detection; DSP Builder; Finite state machine; Energy changing

Received 22 December 2014, Revised 20 January 2015, Accepted 23 January 2015

* Corresponding Author: Lihua Zhao, zhaolh169@163.com

1. Introduction

Speech endpoint detection is to identify non-speech signal and speech signal in noise background environment, and determine the beginning point and ending point. That is considered as one of the key preprocessing components in automatic speech recognition [1]. Recently, there are many endpoint detection methods, such as short-time energy, zero-crossing rate, spectral analysis, spectral entropy, wavelet transform and hidden Markov model (HMM) and so on, many methods are based on software and the calculation is large and complex and therefore can't be performed in real-time detection [2]. But with DSP Builder speech endpoint detection algorithm greatly improves the speed of computation, especially in embedded systems, satisfies hardware environment and the real-time requirements. Therefore, this paper mainly uses the DSP Builder tool to preprocess the input speech signal (pre-emphasis, framing, windows) and using short-time energy speech endpoint detection algorithm [3]. The results show that this method has some validity and feasibility.

2. Speech endpoint detection using short-time energy

2.1. Basic theory of short-time energy to detect the endpoint of speech signal

For noisy speech signal, the energy of the noisy speech segment is sum of the noisy signal energy and speech signal energy [4]. Thus, the energy of the speech segment is larger than the energy of noise segment. At high SNR conditions, by calculating the short-time average energy of the noisy speech signal to distinguish speech segments from background noise. Energy of per frame signal obtained by the following formula:

$$E_j = \sum_{i=0}^{N-1} x^2(i) \quad (1)$$

E_j is the energy of the j-th frame, $x(i)$ is the input speech signal, N is the frame length. For high SNR speech signal, noise energy E_n is small, and the speech signal energy E_n is significantly high, which can be used to distinguish the beginning and ending points of the speech signal.

2.2. Basic theory of short-time energy to detect the endpoint of speech signal using DSP Builder

According to short-time energy during endpoint detection and possible situation of duration, the whole endpoint detection process is divided into 6 states: initial state (S0), the silence state (S1), rising energy state (S2), energy continuous state (S3), decreased energy state (S4) and the energy rise-fall state (S5). Changes depend on its state transition condition (G1 is rising state energy threshold, G2 is continued state energy threshold, E is the current frame energy, Ci is the number of frames in the state i, (1) (2) (3) are priority of the state transition to the next state). Energy state change is shown in Figure 1.

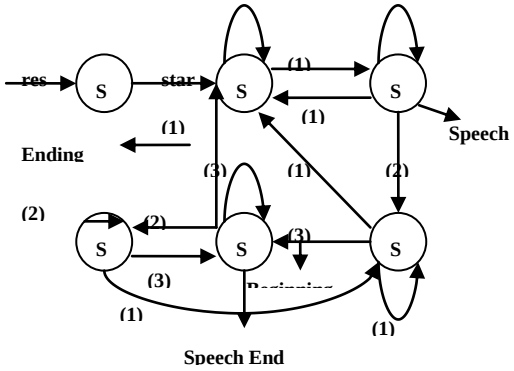


Figure 1. Diagram of the energy state change.

2015 © The American Computational Science Society. All rights reserved.

(1) After the system reset, using the starting 16 frames to calculate the average energy of the noise signal and set it as threshold. Computing the average energy of each frame, obtain the average energy E_{ave} of 16 frames and the maximum frame energy $E_{\max}$, according to $G_i = (E_{\max} - E_{ave}) \times a_i$, i = 1, 2 two dynamical threshold values, where a1 = 3, a2 = 6. G1is starting energy threshold, G2 is sustained energy threshold.

(2) When the start signal is active, enter the S1 state, if one frame energy of speech exceeds the initial energy threshold G1, then enter the S2 state, otherwise stay in the S1 state.

(3) When entering the S2 state, to judge whether more than 3 frames in S2 state.

① If there are more than 3 frames and not enter S3 state, give up the beginning point, that is the pulse.

② If a frame energy exceeds the threshold value G2 then enter S3 state, if maintained for more than 10 frames in the S3 state, it is the beginning point of speech, then enter the S2 state, and its first frame is the start frame, otherwise considered invalid noise, back to S1 state.

(4) After judging the beginning point, then continue to calculate the speech energy, there are two conditions:

① When the speech energy decreased below G1 then enter S4 state, if the number of frames in S4 state more than three, we think the speech is over. And the entering frame of S4 state is the endpoint of the speech.

② If entering S4 state, speech energy is quickly increased higher than G1, then enter the S5 state. In S5 state if the energy is higher than G2 then enters S3 state, if the energy less than G1 then back to S4 state, which is to prevent a small pause in speech judged as speech over.

Effective speech is from the S2 state's first frame to the S4 state's first frame, assuming that the minimum effective length is 15 frames, that is the sum of rising state's frames and continuing state's frames must be more than or equal to 15.

3. Endpoint detection using DSP builder

3.1. Pre-emphasis of the speech signal

Diagram of pre-emphasis model is shown in Figure 2, Delay module is a delay of 1 clock cycle, Barrel Shifter module is a shift register, right shift 4 bit (equal to divide by 2^4).

3.2. Sub-frame process of the speech signal

Diagram of sub-frame process model is shown in Figure 3.

(1)In Figure 3, the input of pre data is the data after pre-emphasis and the output of enframe data is the data after sub-frame. FIFO1 is used for storage data, to control data can be read and written. FIFO2 is variable speed register which can read and output. DMUX is data selector, using sel to control the output data. Signalin is subsystem (shown in Figure 4) which contains mod 256 Counter module and Increment Decrement module.

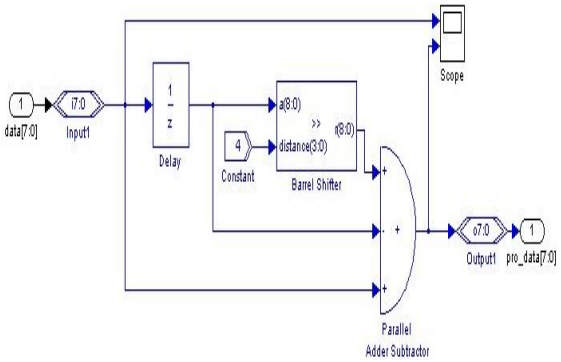


Figure 2. Diagram of pre-emphasis model.

(2) The process of register data: Signalin subsystem's count value is 128 then generate FIFO1 read enable signal, When Counter value is 0 ~127 then reading FIFO1, and writes the read out data to FIFO2, DMUX output FIFO1 data. When Counter value is 128~255 read FIFO2, DMUX output FIFO2 data. After pre-emphasis speech data can be continuously written to FIFO1, real-time dynamic sub-frame speech.

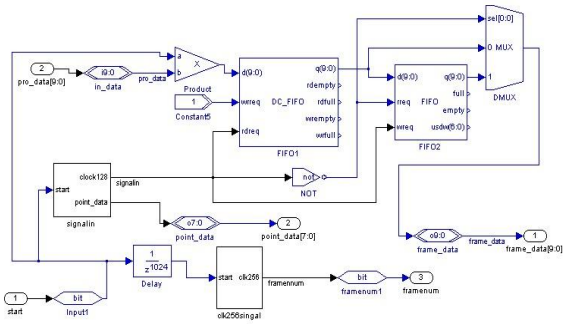


Figure 3. Diagram of sub-frame model.

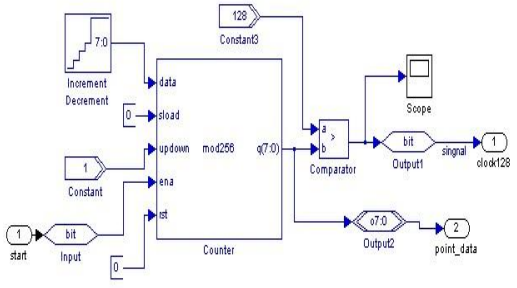


Figure 4. Diagram of Signal in subsystem model.

3.3. Windowing the speech signal and calculating energy of per frame

2015 © The American Computational Science Society. All rights reserved.

Diagram of windowing and calculating energy of per frame is shown in Figure 5.

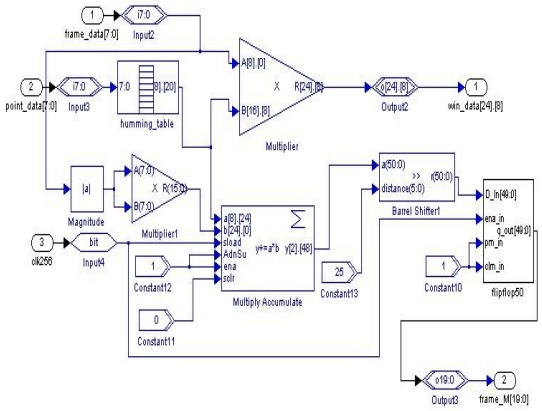


Figure 5. Diagram of windowing and calculating energy of per frame.

We use LUT (look up table) module to realize Hamming window. The LUT input formula is $0.54-0.46*\cos([0:2*pi/(2^8-1):2*pi])$, where n is between 0 and 255. To process the sub-frame and window at the same time, the address is the value of mod 256 counter of sub-frame.

3.4. Recorded speech signal frames

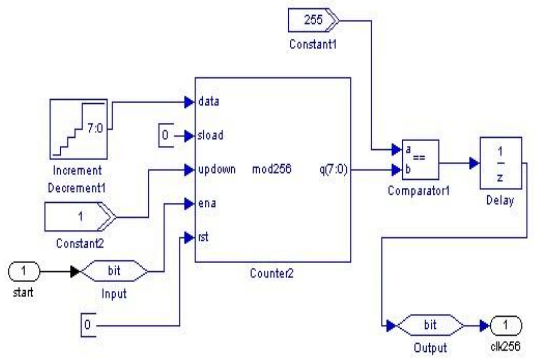


Figure 6. Diagram of subsystem model clk256signal.

In Figure 6 is Clk256signal subsystem which is to produce 256 points periodic pulse signal. Since the speech data is delayed after the sub-frame process, the delay is calculated as about 4 frames that are 1024 points (clock cycles), so join a 1024-point delay in the sub-frame model to synchronizes clk256signal and output of speech data. When clk256signal count 256 points, and then produce a high level, in order to control the recording speech signal frames model.

Diagram of recording speech signal frames is shown in Figure 7.

Counter module enable input signal is output of clk256signal subsystem, counter module is used to record the number of frames, framecounter is output and that is frame number.

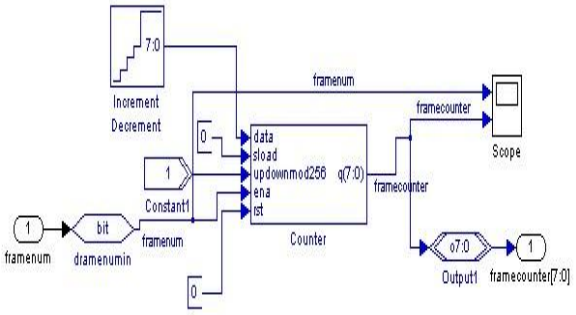


Figure 7. Diagram of recording speech signal frames.

3.5. Setting dual-threshold

Diagram of setting dual-threshold is shown in Figure 8.

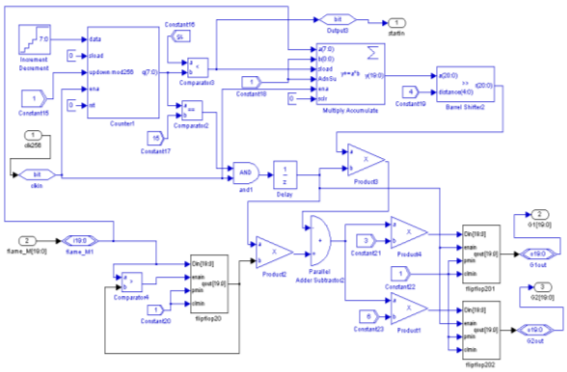


Figure 8. Diagram of setting dual-threshold signal frames.

The processes of design are as follow:

(1)Calculate the average energy: calculate the total energy of the first 16 frames, with a multiply accumulator Multiply Accumulate module to calculate energy of 16 frames, and then divided by 16 to obtain the average energy (Barrel Shifter shift register to the right by 4 bit).

(2) Find the maximum energy: using comparator Comparator4 module to compares the energy of current frame and the energy of last frame, the comparison output is 1 or 0, and that is enable signal of 20 bit D flip-flop. Here E_n is the current frame energy, E_b is the last frame energy, when $E_n > E_b$, the output of D flip-flop is E_n, set E_n as the input of Comparator4 module in order to continue compare, when $E_n < E_b$, D flip-flops keep energy value of last frame, set E_b as the input of Comparator4 module in order to continue compare.

(3)Dual-threshold setting: according to $G_i = (E_{\max} - E_{ave}) \times a_i$, i=1, 2 two dynamical threshold values, where a1 = 3, a2 = 6. G1is starting

2015 © The American Computational Science Society. All rights reserved.

energy threshold, G2 is sustained energy threshold. First calculating the difference between the $E_{\max}$ and E_{ave} by Parallel Adder Subtractior module, and then obtain G1 and G2 by Product module. With D flip-flop to maintain G1, G2 signal.

3.6. FSM design

According to the energy changes and dual-threshold, design FSM with 6 states transition, diagram of FSM model is shown in Figure 9.

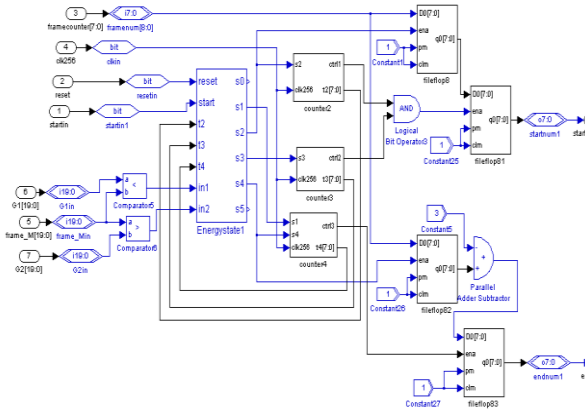


Figure 9. Diagram of FSM and endpoint detection.

In Figure 9, the state machine Energystate1 is State Machine Table of State Machine Functions Library, which can set states and state transition conditions. Reset is a system reset signal, start is starting endpoint detection signal, G1 and G2 are threshold values, the input frame-M is energy of each frame, counter2, counter3 and counter4 are subsystem models, respectively contains a counter to count frame number in state S2, S3andS4, startnum is frame number of beginning point and endnum is frame number of ending point. Where D flip-flop filpflop8, filpflop81, filpflop82, filpflop83 are 8 bit.

3.7. The whole endpoint detection system

The subsystem is integrated into a whole system as shown in Figure 10.

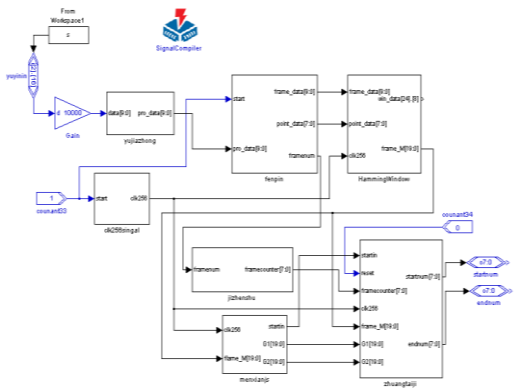
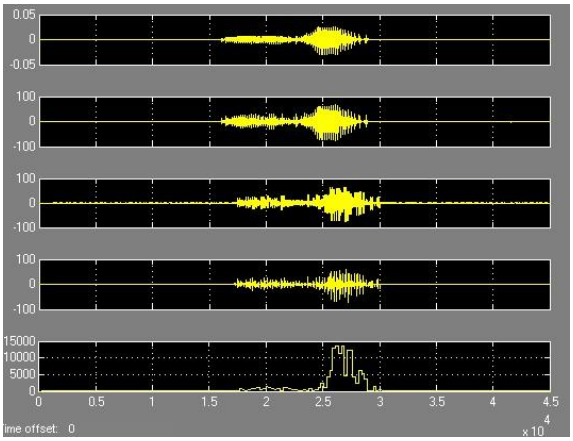


Figure 10. Diagram of the whole system.

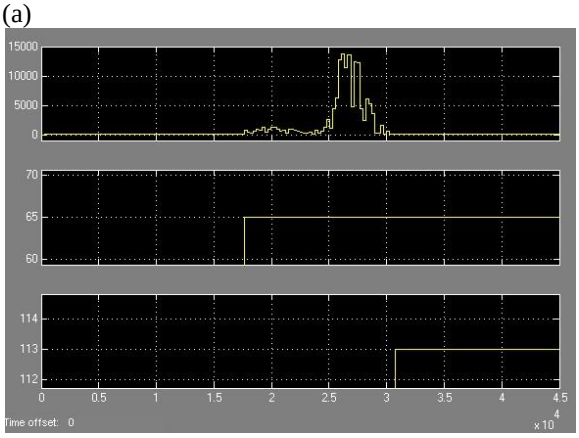
4. Experiment and Results

4.1. The result of endpoint detection using DSP Builder

All of the experimental results are compiled in the DSP Builder software. The speech signal is collected by computer sound card which is wav format. The speech samples using 8 kHz sampling frequency and 8bit quantification, 16 bit sampling precision. The content of the speech signal is Chinese word "1" and "2". We test the effect of endpoint detection.



(a)



(b)

Figure 11. Result of the endpoint detection using DSP Builder.

In order to obtain the result of speech signal process, we add scope model to the DSP Builder to observe the wave of every process. The result is shown in Figure 11.

In Figure (a) is original speech signal wave of Chinese word "one, two", and waves of pre-emphasis, sub-frame, windowed, each frame energy. In Figure (b) is diagram of the energy of each frame and beginning point and ending point. From the Figure (b) we can find the beginning frame number is 65 and the ending frame number is 113. It can detect the endpoint of speech signal clearly.

4.2. Compared with the result of MATLAB software

In order to show the effectiveness of endpoint

2015 © The American Computational Science Society. All rights reserved.

detection using DSP Builder, we compared with the result of endpoint detection using MATLAB software.

Figure 12 is the result of the endpoint detection with short-time energy using MATLAB software. From the figure we can see the beginning point frame is 63 and the ending point frame is 112.

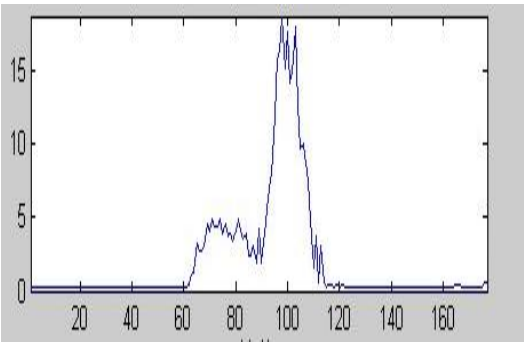


Figure 12. Result of the endpoint detection with short-time energy using MATLAB software.

From the simulation results of endpoint detection using DSP Builder and MATLAB software, we can find that the results DSP Builder endpoint detection is accurately. The DSP Builder model can change into hardware language then programming on FPGA or CPLD. Therefore, this design has a strong practical using.

5. Conclusions

This paper use DSP Builder to realize endpoint detection of speech signal. The results of simulation show the feasibility of endpoint detection using DSP Builder. FSM design the computational complexity and improve the computing speed, moreover, reduces the development time greatly, DSP Builder graphical modeling have advantages which are flexible design, easy to modify and the design process clearly. The ultimate goal of DSP Builder is designed for real-time processing on the hardware chip. Thus this design has good practicability.

Acknowledgment

This work was supported by Northeast Petroleum University and the University-enterprise R & D Center of Measuring and Testing Technology & Instrument and Meter Engineering in Heilongjiang Province.

References

[1] L. H. Zhao, P. Y. Wang, Science Technology and Engineering, 35 (2010) 8822-8825.
[2] L. M. Yan, Y. S. Zhu, J. J. Xu. Transactions of China Electrotechnical Society, 29 (2014) 260-268.
[3] Shorten DSP Design Cycle Using Complex IP in MATLAB/Simulink to Quartus II Flow. Altera Corporation NEWS&Views, 2002.
[4] H. L. Shen, Y. M. Zeng, P. Li, P. Wan, Electron IC Engineer, 134 (2008) 4-6.

2015 © The American Computational Science Society. All rights reserved.