Subtraction procedure for drift and tremor removing from ECG: system level synthesis with Compaan

Tsvetan P. Shoshkov, Georgy S. Mihov

The electrocardiogram (ECG) is often contaminated with interferences such as drift and tremor. The subtraction procedure has proved its efficiency in removing interferences from the ECG. Development process of such device is prone and time consuming. High level synthesis (HLS) tools solve these problems. In this paper HLS tools are used to develop a device for drift and tremor interference removing from ECG. We use Compaan design tool for HLS, which provides full system integration. It automatically generates the HW design and also a SW application to control the HW. The communication between the SW and HW is also automatically utilized through PCIe communication interface. Xilinx Virtex-6 ML605 evaluation kit is used as a target platform. Automatic HW and SW output generation makes the reconfiguration and further development of the device easier and faster. In this paper we make a case study of the used design flow and the implemented design.

Субтракционна процедура за премахване на дрейф и тремор от ЕКГ: Синтез на системно ниво с Компаан (Цветан Шошков, Георги Михов). *Електрокардиограмата (ЕКГ) често е засегната от смущения като дрейф и тремор. Субтракционната процедура е доказала своята ефективност в премахването на смущения от ЕКГ сигнали. Процесът на разработка на такива устройства е труден и изисква много време. Инструментите за синтез на високо ниво решават тези проблеми. В тази статия инструменти за синтез на високо ниво са използвани за разработка на устройство за отстраняване на смущения от дрейф и тремор ит ЕКГ. Използван е Компаан инструмент за синтез на високо ниво, който предоставя възможност за имплементиране на цялостна система. Компаан автоматично генерира хардуерния дизайн и също така софтуерно приложение за управление на хардуера. Комуникацията между софтуера и хардуера е също автоматично осъществена по PCIe интерфейс за комуникация. Xilinx Virtex-6 ML605 се използва като целева платформа. Автоматичното генериране на хардуера и софтуера прави преконфигурирането и развоя на устройството по-лесно и по-бързо. В тази статия правим оценка на използвания процес за разработка и реализираното устройство.*

Introduction

Interferences are often present in the Electrocardiogram (ECG). Such interferences are base-line drift and Electromyographic (EMG) interference. Base-line drift interference occurs because of the complex mechanical and electro-chemical electrode-to-skin processes. EMG interference is present due to involuntary muscle contractions of the patients (tremor). Suppression of such interference is very difficult because its spectrum overlaps the one of the useful signal. Subtraction procedure is proved to be efficient in drift and tremor removing [1], [2], [3], [4], [5].

The basic structure of the procedures for drift and tremor removing are shown on fig. 1 and fig. 2.

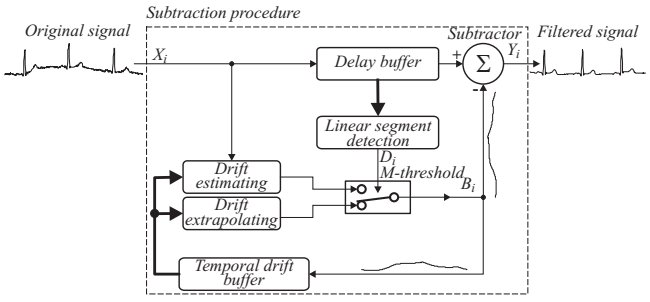

Fig.1. Structure of the Subtraction procedure for drift removing from ECG.

The subtraction procedure algorithms are modeled in Matlab. Based on these algorithms are developed hardware designs for ECG filtering [6], [7]. These designs realize the subtraction procedure for real time

operation. They are developed using VHDL and FPGA as a target platform. This standard manual development process using VHDL is error prone and time consuming. The design often must be reconfigured which is also a difficult process.

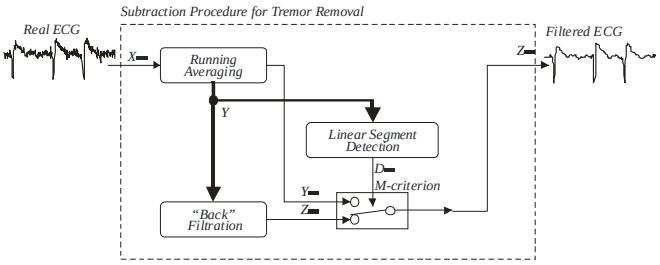

Fig.2. Structure of the Subtraction procedure for tremor removing from ECG.

Several system level development technologies are presented in [9]. Automated high level system generation helps the development process and makes it easier and faster. In the present work we use Compaan to develop our design. Compaan is a high level synthesis tool [8]. Compaan provides full system integration implementing not only the hardware and software implementation but also the communication and integration between them. The design can be easily parameterized and its subsystems can be controlled by automatically generated SW application. Compaan generates not only the SW and HW, but also utilizes the communication between using PCIe interface.

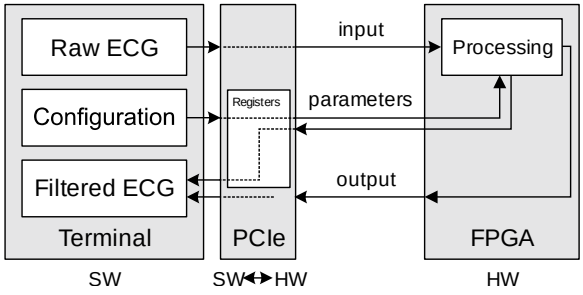

Fig.3. System Level Design with Compaan.

Fig. 3 illustrates the system level design generation using Compaan. The processing block contains the HW design implementing the drift and tremor removing procedures inside the FPGA.

Subtraction procedure for drift removing

The subtraction procedure for tremor removing shown on fig. 1 consists of three basic stages:

- Linear segments detection. Each ECG sample is checked if it belongs to a linear segment by using appropriate linearity criterion. The linearity is defined by comparing the criterion with a predefined threshold M.
- Baseline drift calculation. If it is detected a linear segment the baseline drift is calculated using a digital filter and it is stored in a temporal FIFO buffer. In the same time the baseline drift is subtracted from the linear segment;
- Baseline drift extrapolation. If the current sample belongs to a nonlinear segment the value of the drift is calculated using the data stored in the temporal buffer and it is subtracted from the signal in a nonlinear segment.

Subtraction procedure for tremor removing

The subtraction procedure for tremor removing shown on fig. 2 consists of three basic stages:

– *Filtration of the signal.* Moving averaging is performed for each input token;

– *Detection of linear segments.* Each filtered token from the ECG is defined whether it belongs to a linear segment using an appropriate linear criterion. The calculated linear criterion is compared with a predefined threshold *M*.

– *Back filtration*. If the segment is nonlinear back filtration stage is applied. The current value of the output token is restored using linear-angular interpolation of the signal in the segment.

More information about the subtraction procedure for drift and tremor removing from ECG can be found in [1], [2], [3], [4], [5], [6], [7]. The present paper is focused on the implementation of the Subtraction procedure using the Compaan design flow.

Compaan design flow

Compaan design flow is centered on Compaan compiler and ESPAM tools [8]. Basic stages of the flow are shown on fig. 3.

The input specification must be described as parameterized static affine nested loop which is a subset of the C language. Compaan compiler automatically generates a Khan Process Network (KPN) model of computation based on the input. Next step is to select a target platform and automatically map the resources onto it. HW is automatically generated by ESPAM based on the specifications [8].

Nodes implemented with HW are using processors, which are composed of three separate blocks - read, execute and write. The read block waits until there are tokens ready to be read. A deep pipeline can be integrated in the execution block. Asynchronous work

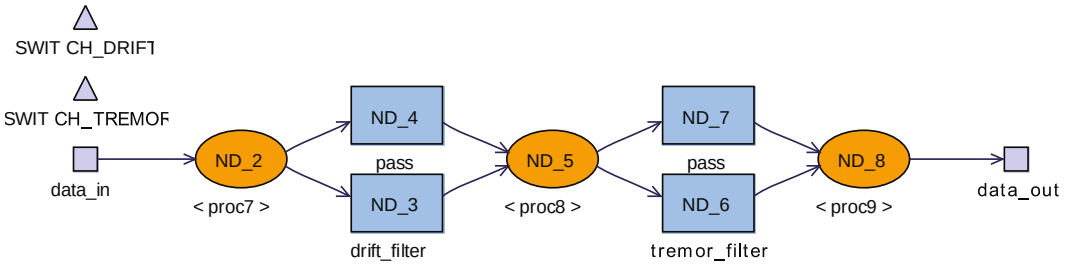

Fig.4. Top Level Subtraction procedure KPN.

of the nodes and automatic links sizes optimizations make the design low power consuming. The high abstraction of the specification and the constant track of the data using counters make the design fault tolerant.

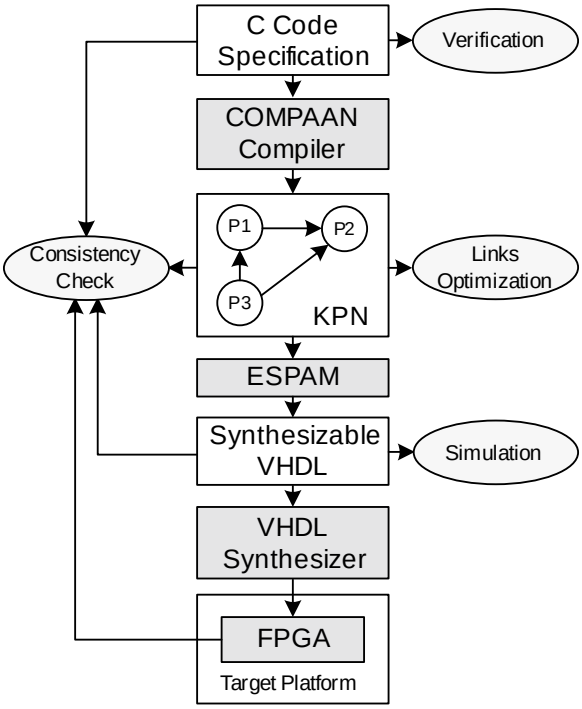

Fig.3. Compaan design flow basic stages.

Implementation the subtraction procedure for drift and tremor removing with Compaan

The basic structure and elements of the Subtraction Procedure for drift and tremor removing are shown on fig. 1 and fig. 2. We use the Compaan design flow which is shown on fig. 3 to implement the subtraction procedure. The following sections give detailed description for each step of the design flow.

Input specification

As a first step of the development it is created the input SW specification. Short fraction of the C code input specification is presented to picture its main concepts:

```
#pragma compaan_procedure ecg_filter
void ecg_filter(int data_in[WIDTH], int
data_out[WIDTH]) {
  int ecg[WIDTH];
  int drift_out[WIDTH];
  int tremor_out[WIDTH];
  int tremor_in[WIDTH];
  int i, j, x;
  Parameters *pr[1];

  #pragma compaan_parameter 0 1
  const unsigned int SWITCH_DRIFT =
parameters->sw_drift;
  #pragma compaan_parameter 0 1
  const unsigned int SWITCH_TREMOR =
parameters->sw_tremor;
  producer(pr[0]);
  // Stream data into the design
  for (i = 0; i < WIDTH; i = i + 1) {
    ecg[i] = data_in[i];
  }
  // Hierarchical call
  if(SWITCH_DRIFT <= 0){
    drift_filter(ecg, &drift_out);
  } else {
    pass(ecg, &drift_out);
  }
  for (j = 0; j < WIDTH; j++) {
    tremor_in[j] = drift_out[j];
  }
  // Hierarchical call
  if(SWITCH_TREMOR <= 0){
    tremor_filter(tremor_in, &tremor_out);
  } else {
    pass(tremor_in, &tremor_out);
  }
  // Stream data out
  for (x = 0; x < WIDTH; x = x + 1) {
    data_out[x] = tremor_out[x];
  }
}
```

Functions are called inside static affine nested loops to process the data. Inputs and outputs of the functions are given as parameters. Output parameters are specified as addresses to variables where the result is stored.

Parameters are used to turn on and off the procedures for drift and tremor removing. Here we present the static affined nested loop which is part of the *drift_filter* function:

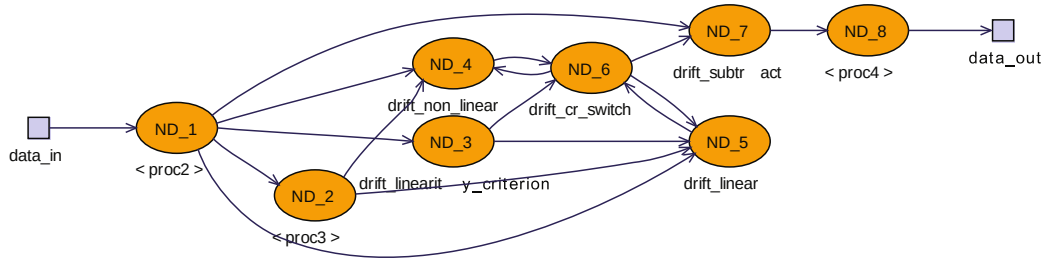

Fig. 5. Drift Removing KPN.

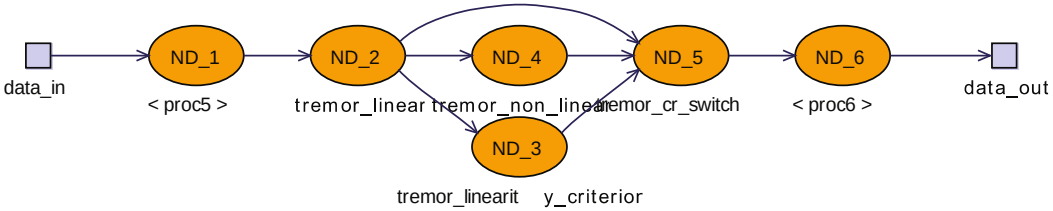

Fig. 6. Tremor Removing KPN.

```
for (j = 1; j <= WIDTH; j = j + 1) {
  drift_linearity_criterion(ecg[j], &cr[j],
&cfr[j]);
  drift_non_linear(ecg[j], ecg_drift[j-1],
&ecg_out_non_linear[j]);
  drift_linear(ecg[j], ecg_drift[j-1],
cfr[j], &ecg_out_linear[j]);
  drift_cr_switch(cr[j], ecg_out_linear[j],
ecg_out_non_linear[j], &ecg_drift[j]);
  drift_subtract(ecg[j], ecg_drift[j],
&ecg_filtered[j]);
}
```

There are loops to stream in and out data located before and after the given part of the drift_filter function. The data processing for drift removing starts with the first block witch is linearity criterion calculation. Functions for linear and nonlinear segment filtrations are present. Another function switches between the nonlinear and linear output based on the linearity criterion. The drift interference is subtracted by the original ECG signal.

Here is a fraction of the tremor_filter function implementing the tremor removing:

```
for (j = 1; j <= WIDTH; j = j + 1) {
  tremor_linear(ecg[j], &filtered[j],
&ecg_linear[j]);
  tremor_linearity_criterion(filtered[j],
&cr[j]);
  tremor_non_linear(filtered[j],
&ecg_non_linear[j]);
  tremor_cr_switch(cr[j], ecg_linear[j],
ecg_non_linear[j], &ecg_filtered[j]);
}
```

The data processing for tremor removing starts with the first block witch is filtering the input signal. Linearity criterion is calculated based on the filtered signal. Another function implements the back filtration for the nonlinear segments. Nonlinear and linear outputs are switched based on the linearity criterion.

KPN Model of computation

Compaan compiler is used to automatically generate KPN specification. This specification can be used as a basis to generate the actual implementation of the device. The KPN defines the dependencies between the nodes and connections between them.

The KPN is simulated to automatically optimize the communication link sizes. Fig. 4 shows the produced top level KPN for the Subtraction Procedure. Each function from the input specification is realized using a separate processing node. Additional processing nodes are used to stream data in and out from the design. It is used a hierarchical structure. Subnetwork KPNs are generated for the tremor and drift removing procedures. Fig. 5 and fig. 6 show the drift and tremor removing KPN subnetworks.

Implementation

Next step of the Compaan design flow is selecting target platform and mapping creation. In our case we use default target and mappings, since we want to generate a project to simulate. After we have the KPN, platform and mapping specifications ready, we can generate synthesizable VHDL code. Each node is mapped to a hardware processing core. It consists of read, execute and write section. Each execution section must be additionally filled with the actual processing equations. The manually filled code realizes the relation between the inputs and the outputs in each block. An alternative in Compaan is to

use third party tools to automatically generate the hardware logic. Also processor cores can be used to implement the processing algorithms.

At each step of the development flow we perform consistency check with the other stages to define if there are any errors. Thus errors are easy to be identified and fixed at earlier stages of the development.

Evaluation and Results

Main target of our evaluation is to check if the generated design satisfies the application requirements. The development time is also a major aspect in this research. The system design must meet the functional and performance requirements and at the same time must be low power consuming and fault tolerant.

We use Xilinx ISE Design Suite 14.7 to simulate and verify the design. Compaan automatically generates a test bench that streams in and out data from the design. We use ECG signal provided by the American Health Association to test our design. Finally we use Matlab to plot the signals in graphics.

Fig. 7 shows result signals which prove that the integrated design realizes the subtraction procedure according to the modeled algorithms.

Manual VHDL implementation of the subtraction procedure for tremor removing is presented in [6]. We can compare it with developing the tremor removing using Compaan design flow. Table 1 presents a comparison between the manual and automatic design generation for tremor removing.

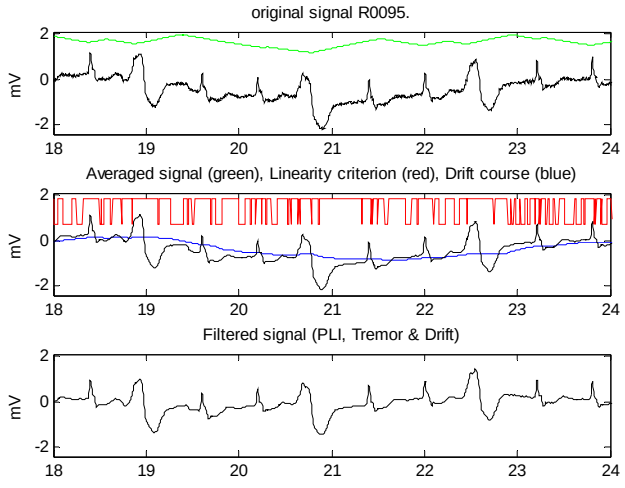

Fig.7. Drift and tremor removing result signals.

Using Compaan the development time required is considerably decreased and the occupation of the FPGA is relatively the same.

Table 1

Subtraction Procedure for Tremor Removing Development Methods Comparison.

Development Method	Target Platform	Slices Occupied	Man-Months
Manual	Spartan 3	815 (7%)	1
Compaan	Virtex 6	700 (1%)	0.4

The improvement in the development time is achieved thanks to the automation of the code generation and optimization. Using Compaan errors can be detected in earlier stages of the development. Hardware processing blocks and communication links are automatically created.

Conclusions

We use high level synthesis tool to develop a device that implements the subtraction procedure for drift and tremor removing from ECG. Compaan design flow automatically generates an output design based on simple input specification. This considerably decreases the development time required. Compaan further optimizes the design and makes it fault tolerant and low power consuming. Quick changes in the specification can be made to generate a new design. SW application is generated and the communication between the SW and HW through PCIe is additionally utilized. This benefits any development and research process.

Acknowledgements

This paper is a part of a project under contract № 142ПД0040-03/2014, which is funded by the research program of the TU – Sofia, Bulgaria.

REFERENCES

[1] Dotsinsky, I., G. Mihov. Tremor Suppression in the Electrocardiogram. Electronics ET2007, Proceedings of the Sixteenth International Scientific and Applied Science Conference, b. 2, Sozopol, 2007, pages 19-26.

[2] Dotsinsky, I., G. Mihov. Tremor Suppression in ECG. BioMedical Engineering OnLine, 7:29, 2008.

[3] Dotsinsky, I., G. Mihov. Simple Approach for Tremor Suppression in Electrocardiograms. International Journal BIOautomation, 14(2), 2010, pages 129-138.

[4] Shoshkov, T. Subtraction Procedure for ECG Drift Removing: High Level Synthesis with Compaan. ICEST, Ohrid, vol 1, pp. 241-244, 2014

[5] Mihov, G., I. Dotsinsky. Subtraction Procedure For Removing The Baseline Drift From ECG Signals. Annual Journal of Electronics, Vol. 4 Issue 2, pp. 118-122, 2010.

[6] Shoshkov, T., G. Mihov. Субтракционна процедура за премахване на ЕМГ смущения от ЕКГ сигнали: Адаптация за работа в реално време с програмируеми устройства. Electronica, Sofia, 2013, pp 73-78

[7] Shoshkov, T., G. Mihov. Subtraction Procedure for Removing the Baseline Drift from ECG Signals: Adaptation For Real Time Operation With Programmable Devices. ICEST, Ohrid, 2013, vol 1, pp. 687-690

[8] Stefanov, T., C. Zissulescu, A. Turjan, B. Kienhuis, Ed Deprettere. System Design using Khan Process Networks: The Compaan/Laura Approach. DATE, Paris, pp. 340-345, 2004

[9] Gerstlauer, A., C. Haubelt, A. Pimentel, T. Stefanov, D. Gajski, J. Teich. Electronic System-Level Synthesis Methodologies. IEEE Transactions on Computer-Aided Design of Integrated Circuits, vol. 28, No. 10, pp. 1517-1530, Oct 2009

M.Sc. Eng. Tsvetan P. Shoshkov *is with the Faculty of Electronics, Department of Electronics, TU – Sofia, Bulgaria. Interests are mainly in the field of digital electronics, FPGA and high level synthesis.*

tel.: 0888 565 278 *e-mail: tsh@tu-sofia.bg*

Prof. D.Sc. Eng. Georgy S. Mihov *is with the Faculty of Electronics, Department of Electronics, TU – Sofia, Bulgaria. Interests are mainly in the field of digital electronics, microprocessor system application, digital filtration.*

tel. (02) 965 2560 *e-mail: gsm@tu-sofia.bg*

Received on: 15.07.2014