Heart-on-a-Chip: A Closed-loop Testing Platform for Implantable Pacemakers

Zhihao Jiang  
*University of Pennsylvania*, zhihaoj@seas.upenn.edu

Sriram Radhakrishnan  
*University of Pennsylvania*, sriramr@seas.upenn.edu

Varun Sampath  
*University of Pennsylvania*, vsampath@seas.upenn.edu

Shilpa Sarode  
*University of Pennsylvania*, sarode@seas.upenn.edu

Rahul Mangharam  
*University of Pennsylvania*, rahulm@seas.upenn.edu

Follow this and additional works at: [http://repository.upenn.edu/mlab_papers](http://repository.upenn.edu/mlab_papers)

Part of the [Computer Engineering Commons](http://repository.upenn.edu/mlab_papers)

**Recommended Citation**


This paper is posted at ScholarlyCommons. [http://repository.upenn.edu/mlab_papers/55](http://repository.upenn.edu/mlab_papers/55)
For more information, please contact libraryrepository@pobox.upenn.edu.
Heart-on-a-Chip: A Closed-loop Testing Platform for Implantable Pacemakers

Abstract
Implantable cardiac pacemakers restore normal heart rhythm by delivering external electrical pacing to the heart. The pacemaker software is life-critical as the timing of the pulses determine its ability to control the heart rate. Recalls due to software issues have been on the rise with the increasing complexity of pacing algorithms. Open-loop testing remains the primary approach to evaluate the safety of pacemaker software. While this tests how the pacemaker responds to stimulus, it cannot reveal pacemaker malfunctions which drive the heart into an unsafe state over multiple cycles. To evaluate the safety and efficacy of pacemaker software we have developed a heart model to generate different heart conditions and interact with real pacemakers. In this paper, we introduce the closed-loop testing platform which consists of a programmable hardware implementation of the heart that can interact with a commercial pacemaker in closed-loop. The heart-on-a-chip implementation is automatically generated from the Virtual Heart Model in Simulink which models different heart conditions. We describe a case study of Endless Loop Tachycardia to demonstrate potential closed-loop pacemaker malfunctions which inappropriately increase the heart rate. The test platform is part of our model-based design framework for verification and testing of medical devices with the patient-in-the-loop.

Keywords
FPGA, Medical Device Testing

Disciplines
Computer Engineering

This technical report is available at ScholarlyCommons: http://repository.upenn.edu/mlab_papers/55
Heart-on-a-Chip: A Closed-loop Testing Platform for Implantable Pacemakers

Zhihao Jiang, Sriram Radhakrishnan, Varun Sampath, Shilpa Sarode, Rahul Mangharam
Department of Electrical and System Engineering
University of Pennsylvania
{zhihaoj, sriramr, vsampath, sarode, rahulm}@seas.upenn.edu

Abstract—Implantable cardiac pacemakers restore normal heart rhythm by delivering external electrical pacing to the heart. The pacemaker software is life-critical as the timing of the pulses determine its ability to control the heart rate. Recalls due to software issues have been on the rise with the increasing complexity of pacing algorithms. Open-loop testing remains the primary approach to evaluate the safety of pacemaker software. While this tests how the pacemaker responds to stimulus, it cannot reveal pacemaker malfunctions which drive the heart into an unsafe state over multiple cycles. To evaluate the safety and efficacy of pacemaker software we have developed a heart model to generate different heart conditions and interact with real pacemakers. In this paper, we introduce the closed-loop testing platform which consists of a programmable hardware implementation of the heart that can interact with a commercial pacemaker in closed-loop. The heart-on-a-chip implementation is automatically generated from the Virtual Heart Model in Simulink which models different heart conditions. We describe a case study of Endless Loop Tachycardia to demonstrate potential closed-loop pacemaker malfunctions which inappropriately increase the heart rate. The test platform is part of our model-based design framework for verification and testing of medical devices with the patient-in-the-loop. 1

I. INTRODUCTION

The Electrical Conduction System of the heart coordinates the contraction of heart muscles for efficient blood circulation within our body. Derangements in the rhythm of generation and conduction of electrical signals within the heart tissue result in abnormal heart rhythm, which is referred to as an arrhythmia. Rhythm management devices, such as the implantable cardiac pacemaker, are designed to sense electrical activity of the heart, diagnose the current heart condition, and deliver artificial pacing therapies to maintain appropriate heart rhythm. The direct interaction between the pacemaker and the heart makes the closed-loop system a perfect example of a Cyber-Physical System and the safety of the device operation is life-critical.

During the past decade, over 600,000 cardiac devices were recalled due to flaws which caused adverse health consequences or death [1]. With increasing complexity of device software, recalls are due to software-related malfunctions [2] have been rising. While the US Food and Drug Administration (FDA) require device manufacturers to establish the safety and efficacy of their devices, they do not explicitly specify the methods to verify the device software design or analyze the code [3].

Platform testing remains the primary means to verify and validate device software. Currently testing is done by feeding recorded open-loop heart signals to the device and evaluating the device output. However, the change in the state of the heart condition, in response to device output, is not taken into account. Thus, device malfunctions involving state changes due to multiple closed-loop interactions will not be captured during testing.

The heart-on-a-chip platform Fig. 1 utilizes the model-based framework for developing safe device software proposed in [4]. The outline of the framework is shown in Fig. 2; at verification level, the safety of the Boston Scientific pacemaker specification is first evaluated by verifying the abstract model of the algorithms in the UPPAAL model checker [5]. The verified model of the pacemaker is then automatically translated into Stateflow/Simulink using the UPP2SF tool [4]. At the simulation level, safety violations from verification level, as well as other non-timing aspects of the pacemaker are checked against more detailed heart model. Using the Simulink Coder, the Stateflow model of the pacemaker is translated into C code. At testing level, hardware-related aspects (Scheduling delays, electrical interference, etc.) of the pacemaker can be evaluated. This end-to-end tool chain guarantees the verified properties are preserved during model translation and code generation which can save significant debugging time.

At each level of the tool chain, it is important to have a model of the heart which is able to represent different heart

---

1This research work was supported in part by the NSF CAREER 1253842 NSF CPS-0931239 and MRI-0923518 grants.
conditions and responds to device outputs. The model should be implemented in hardware to interact with both white-box and black-box (commercial) pacemaker implementations. To this effect, Jiang et al. developed the Virtual Heart Model (VHM) to model the electrical conduction system of the heart [6]. The goal of this effort is to demonstrate the interaction of this heart model on a programmable hardware platform with a commercial pacemaker, and illustrate the importance of closed-loop testing with a clinically relevant case-study.

The remainder of the paper is organized as follows: Section 2 introduces the Virtual Heart Model (VHM) and its Simulink implementation. Section 3 describes how the simulink model is automatically translated into Verilog code using Simulink Hardware Description Language (HDL) Coder [7] and operated on a Field Programmable Gate Array (FPGA) programmable hardware platform [8]. Through an analog interface the platform, the heart-on-a-chip is able to interact with a Boston Scientific pacemaker [9]. In Section 4, we demonstrate the use of the platform in closed-loop testing. We use Endless-Loop Tachycardia (ELT) as an example to demonstrate the necessity of closed-loop testing. ELT is a closed-loop execution where the pacemaker increases the heart rate inappropriately. This behavior would not arise during open-loop testing which serves as a perfect example for the necessity of closed-loop testing.

In our previous work [6], we manually implemented the VHM model on a FPGA chip and a pacemaker model on a microcontroller, and showed several closed-loop behaviors through a digital interface. This work extends our previous work in three aspects: (a) the heart model implementation process is automated, which is a key process in a model-based design framework; (b) With the analog interface the testing platform is capable of interacting with a commercial pacemaker. This makes the platform a suitable tool for black-box testing and during pacemaker certification process. (c) We use the platform to demonstrated the necessity of closed-loop testing by triggering the Endless-Loop Tachycardia (ELT) on a real pacemaker.

II. VIRTUAL HEART MODEL (VHM)

A. Timing properties of the heart tissue

When the electrical potential outside of a heart muscle change, a series of ion channel activities will happen on the membrane of the cell and the heart muscle will contract. The resulting change of the electrical potential outside the tissue is referred to as Action Potential. (Fig. 4 (b)) The sudden increase of the potential is referred to as Depolarization. This potential change will then depolarize adjacent heart tissue, creating a activation wave throughout the heart. The time needed for the potential to reach depolarization threshold is the conduction delay between tissues. After the depolarization the tissue has to recover from ion depletion. The timing period before the action potential drop to resting potential is referred to as Refractory period and can be divided into Effective Refractory Period (ERP) and Relative Refractory Period (RRP) (Fig. 4 (b)). The tissue responds to depolarization wave differently during the refractory periods. During ERP the depolarization wave is blocked due to lack of ion. (Fig. 4 (c1)) During RRP the conduction delay of the tissue is increased due to the increased slope of the action potential. (Fig. 4 (c2)) Heart tissue with different refractory and conduction periods form the electrical conduction system of the heart (Fig. 3 (a)) and governs the coordinated contractions of heart muscle for optimal blood flow.

Heart tissue can also be depolarized by artificial electrical signals. Implantable cardiac devices like pacemaker monitor local electrical activities of the heart by inserting leads into the heart and deliver electrical pacing to restore normal heart rhythm.
The heart can be viewed as a conduction network consists of conduction paths. In VHM, we use node automata (shown in Fig. 5(a)) to model the refractory property of heart tissue. In each state the node respond to external activation differently. During ERP the node cannot be activated thus no signal can conduct through. During RRP the signal will be conducted through with increased delay. We use path automata (Fig. 5(b)) to abstract the conduction delay of the heart tissue between nodes. The path can conduct in both directions (Ante and Retro) and activate the node on the other end after timer runs out. The heart is modeled by connecting a set of node and path automata into a conduction network (Fig. 3(b)). Both node and path automata are implemented in Simulink and the path automata into a conduction network (Fig. 3(b)).

B. Timing Model of the Heart

The VHM, as well as the monitoring system, are automatically translated into Verilog using Simulink HDL coder. Machine-generated code tends to be very conservative, as the generator will prefer to unambiguously maintain model semantics over any possible performance optimizations. The machine-generated code then may implement additional logic and/or registers that may be unnecessary for the desired computation. In the worst case, the machine-generated code might implement computations which maybe infeasible on the FPGA, since the generator does not know the exact specifications of the underlying hardware. As a result, generated HDL will place a greater burden on FPGA capacity constraints. The trade-off with automated code generation is the lack of optimization that handwritten code will have, but preserves the properties of the model and save design time. In this project we are not sacrificing model functionality in order to fit these constraints. Instead we optimize the models in order to generate more efficient HDL. In other words, by redesigning the models to use constructs that we know will synthesize correctly and efficiently on the FPGA. In the heart model implementation considered here, the Verilog code generated from a VHM with 19 nodes and 19 paths consumed 13,319 look-up tables (LUTs) on the DE0-Nano FPGA platform [8], accounting for less than 11% of the available resources. With a Verilog wrapper which maps the inputs and outputs to general-purpose I/O (GPIO) pins, FPGA chip is able to send and sense digital events to and from external pacemaker devices.

B. Analog Interface with the Pacemaker

In order to interact with a commercial pacemaker, an analog interface is needed for signal conditioning. The pacemaker uses bi-polar leads to both sense and pace the heart. The circuit diagram is shown in Fig. ?? Our analog interface uses an optical isolation circuit to separate the pacemaker circuit and the heart implementation. Signals generated form the heart are attenuated to the appropriate level to interact wit the Boston Scientific pacemaker and analog pacing signals are converted to pacing events received by the heart model.

C. Programmable user interface

An user interface has been developed in Matlab (Fig. ??). Information such as activations of node automata and requirement satisfaction are sent to the front-end through a serial port. The operator can change VHM parameters at runtime and safety violations issue notifications.

IV. CLOSED-LOOP TESTING

We briefly describe the basic timing functions of a pacemaker and a case-study of closed-loop testing of a commercial pacemaker.
A dual chamber pacemaker senses the electrical activation of the atria and ventricles with two leads inserted into the right atrium and ventricle, respectively. The primary function of a dual chamber pacemaker is to 1) maintain an adequate ventricular rate, 2) maintain the appropriate Atrial-Ventricular delay. The five basic timing cycles for a dual chamber pacemaker is to 1) maintain an adequate ventricle and atrium, respectively. The primary function of a dual chamber pacemaker. More detailed descriptions of the cases and safety invariants evaluated are in [10].

A. Basic Timing Cycles for a Dual Chamber Pacemaker

A dual chamber pacemaker senses the electrical activation of the atria and ventricles with two leads inserted into the right atrium and ventricle, respectively. The primary function of a dual chamber pacemaker is to 1) maintain an adequate ventricular rate, 2) maintain the appropriate Atrial-Ventricular delay. The five basic timing cycles for a dual chamber pacemaker is shown in Fig. 6(a). After each sensed (VS) or paced (VP) ventricular event, the pacemaker will deliver Atrial Pacing (AP) if no Atrial Sense (AS) event occurs within TAEI. If no ventricular events has been sensed (VS) with in TAVI after every sensed (AS) or paced (AP) atrial event, the pacemaker will deliver ventricular pacing (VP). The pacemaker also has a Upper Rate Interval (URI) timer to track the interval between two ventricular events. If the URI period is not finished, the ventricular pacing (VP) is delivered after the period finishes. The pacemaker also has Post Ventricular Atrial Refractory Period (PVARP) and Ventricular Refractory Period (VRP) to mimic the signal blocking of the heart tissue in order to filter noise.

B. Endless Loop Tachycardia (ELT)

The pacemaker maintains a maximum interval between an atrial event (AS, AP) and a ventricular event (VS, VP), which is equivalent to a virtual conduction pathway between the atria and the ventricles. Along with the intrinsic A-V conduction pathway (i.e. heart tissue), the pacemaker in the loop represents a conduction loop along which electrical signal information can travel through (Fig. 6(a)). While the normal rhythm of the heart is the common case, a not so uncommon scenario in the heart is a Premature Ventricular Contraction (PVC), in which the ventricles contracts prematurely and spontaneously. In the closed-loop setting (Fig. 6(b) (a)), when a PVC occurs within a timing window (shown with Marker 1 in Fig. 6(b)), it will trigger electrical conduction from the ventricles to the atria through the intrinsic pathway. The pacemaker registers this signal as an Atrial Sense (AS) (Marker 2 in Fig. 6(b)). This AS event triggers VP after TAVI, as if the signal conducts through the virtual A-V pathway (Marker 3 in Fig. 6(b)). The VP will trigger another V-A conduction and the signal is traveling within the loop. Since the cycle of the loop is faster than intrinsic heart rate, this VP-AS-VP-AS pattern will persist and inappropriately increase the heart rate, and this closed-loop pacemaker malfunction is referred to as Endless-Loop Tachycardia (ELT).

ELT is a closed-loop interaction between the heart and a dual chamber pacemaker. Without the heart responding to pacemaker pacing, ELT will not occur and thus cannot be found during open-loop testing. In open-loop setting (Fig. 6(b) (b)), assuming the same heart condition, the PVC can also trigger a V-A conduction (Marker 4 in Fig. 6(b)), which will reset the intrinsic atrial event. However, without taking into account the pacemaker pacing, the conduction loop does not exist and no more V-A conduction will be triggered as in the closed-loop case (Marker 5 in Fig. 6(b)). By examining the pacemaker pacing (dashed VP events in Fig. 6(b) (b)) after feeding the open-loop signal as input to the pacemaker, we may conclude that the pacemaker is operating correctly, and miss the opportunity to discover the closed-loop malfunction.

C. Identify ELT Using Our Testing Platform

In our testing platform, the PVC event can be triggered either randomly by the platform or by user input. When the interval between the last ventricular event (VS, VP) to the PVC is longer than the blocking period of ventricular tissue, a V-A conduction can be triggered thus trigger ELT when connected to a dual chamber pacemaker. We have successfully triggered ELT execution within the closed-loop system consists of the testing platform and a Boston Scientific pacemaker.

V. Conclusion

In this paper, we presented Heart-on-a-Chip, a closed-loop testing platform for interactive testing of implantable pacemakers. We introduced the automated code generation of the Virtual Heart Model (VHM) from Simulink model to HDL code and its implementation on a FPGA-based processor. Through an analog interface circuit the heart model implementation can interact with a commercial pacemaker. The testing platform is able to represent different heart conditions for closed-loop testing of implantable pacemaker. We use the Endless-Loop Tachycardia, which is a clinical closed-loop pacemaker malfunction that the pacemaker inappropriately increases heart rate, as case study to show the necessity of closed-loop testing.

REFERENCES


