Compsys 303 Assignment 1 Report
Team 10

Rachel Nataatmadja

rnat697

980468406

Shou Miyamoto

smiy200

168133060

Introduction
In this project, we are designing a DDD mode pacemaker in SCCharts and implementing it to a Nios II CPU, DE2-115. On the DE2-115, two modes must be available to use: Mode 1 and Mode 2. Mode 1 will use KEY0 and KEY1 to simulate atrial and ventricular events from the heart. If there are no inputs from KEY0 or KEY1, it should pace the respective atrial or ventricular pulse which is indicated on the green LEDs. Mode 2 will allow the pacemaker to communicate over UART to a virtual heart running on an exe file on the PC. Any missing atrial or ventricular event will cause the pacemaker to pace the respective event over the UART.

Methodology
As team 10 we approached this assignment given via a top down approach. We first examined the inputs and outputs and listed them as such:

Inputs:
Inputs are defined as signals that are emitted from an external source that can not be determined.

- AS, Atrial Sense inputted as a signal
- VS, Ventricle Sense inputted as a signal

Outputs:
Outputs are determined by deadlings that are missed and need to be outputted in order to create a functioning heart system.

- AP, Atrial Pace,a signal outputted when a timing deadline is missed for AS
- VP, Ventricle pace, a signal outputted when a timing deadline is missed for a VS

The next stage we considered to examine is the state of timers throughout the pacing of the pacemaker. From SSCharts we created 6 timing deadlines and outlined what was required of each deadline.

Timers/Counters
The maximum timeouts that could occur for each critical state was described and set accordingly in the timer.h header file.

Interrupts/State transition
We set up interrupts in buttons pushed on mode 1 and encapsulated our code in two conditionals that constantly check which mode we would be in according to the switch cases.

As a team we decided to only use the two far right switches (SW0 and SW1) that account for the binary bits of 0 and 1 because this seemed like a logical route to go.

Mode 1 would be when the furthest right would be on, (0,1) and Mode 2 would be when the second switch, (1,0) would be on. In the case both two switches would be on we prioritised the second bit as opposed to the first.

States
AVI - Atrioventricular Interval:
This is a deadline describing the maximum time between an atrial and its subsequent ventricular event. This means that the timing deadline starts at an AS or AP signal until a VS signal occurs.

If no VS signal occurs during this interval and the maximum time has been reached, a VP signal is output which paces the ventricular.

In addition, if the AVI finishes before the duration specified for URI, the AVI is extended until the end of the URI specified. A VP would then be outputted provided VS has not occurred.

PVARP - Post-Ventricular Atrial Refractory Period:
The time after a ventricular event where any atrial events can be ignored as AR (atrial refractory) signals. This time

expires when it reaches the designated time set by an external source in timing.h

VRP – Ventricular Refractory Period:
The time after a ventricular event where any subsequent ventricular events can be ignored as VR (ventricular refractory) signals. This time expires when the designated time set by the header file has been reached.

AEI– Atrial Escape Interval:
The maximum time between a ventricular event and the subsequent atrial event. This timing deadline triggers on a VS or VP signal until an AS has occurred. If the maximum time has been reached and an AS has not occurred, an AP signal is outputted which paces the atrial.

LRI– Lower Rate Interval:
The slowest rate at which the heart can operate and is measured by the time between ventricular events. The time can be reset if another VS or VP signal is seen earlier than the expected time.

URI – Upper Rate Interval:
The fastest rate at which the heart can operate and is measured by the time between ventricular events. This time is fixed.

Results
There are two primary methods that were used to verify our results. The first being a tool provided by Nathan, https://us.nallen.me/cs303/, which is a tool to test out our SCCharts implementation. The second tool would be the CS302-heart.exe file that simulates a virtual heart on a PC over UART.

We utilised our first test tool after creating and templating all critical timers to verify each situation was functioning as intended.

Once confirmed that all timers were functioning. The project was then ported over to Quartus Prime's Nios II (Eclipse) where we started to explore the transition of mode 1 and 2 using switches and the usage of KEY0 and KEY1.

The following figures 1, 2, 3, 4 and 5 are graphs that were taken from the given tool for SCCharts testing. Any positive values on the graphs shown indicate natural atrial or ventricular events. Any negative values on the graphs indicate external stimulation from the pacemaker on the corresponding atrial or ventricular events.

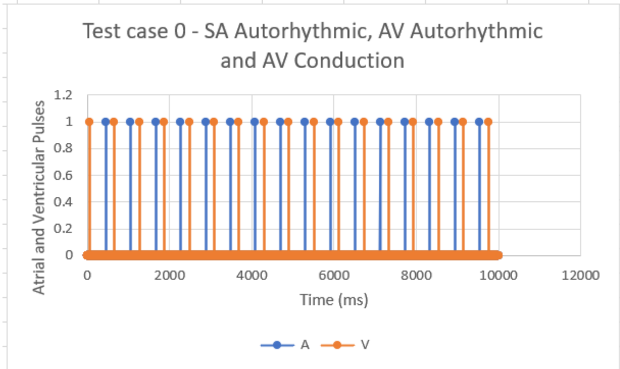

Fig 1. Graph of Test Case 0 - All Natural Heartbeat

In figure 1, we expect no negative pulses shown in the graph. This is because the heartbeat has a normal natural pacing. Therefore, no atrial or ventricular pacing from the pacemaker is required for this.

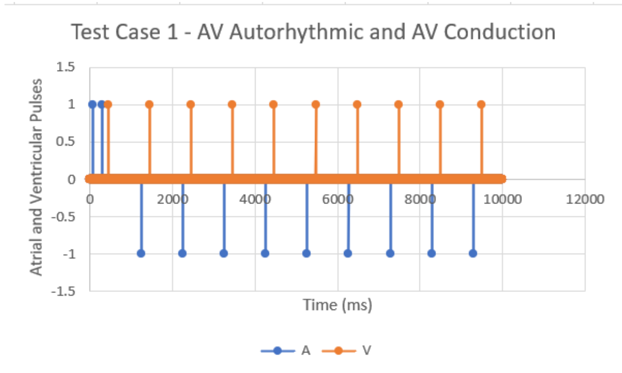

Fig 2. Graph of Test Case 1 - AV Autorhythmic and AV Conduction

In figure 2, we expect negative pulses coming from the atrial on the graph. This is because there are no natural atrial pulses coming from the heart. Therefore, our design should pace the atrial, which it does.

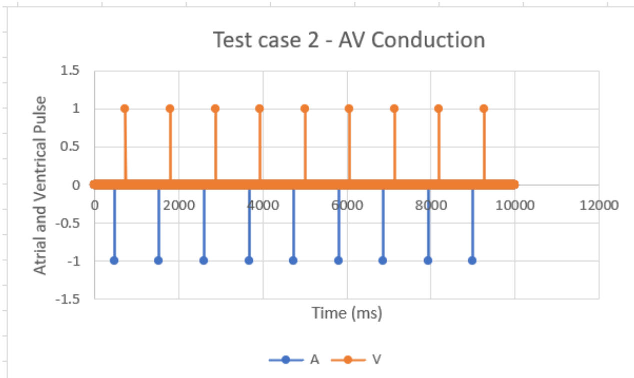

Fig 3. Graph of Test 2 - AV Conduction

Similar to figure 2, figure 3 also requires atrial pacing from the pacemaker due to missing natural atrial events.

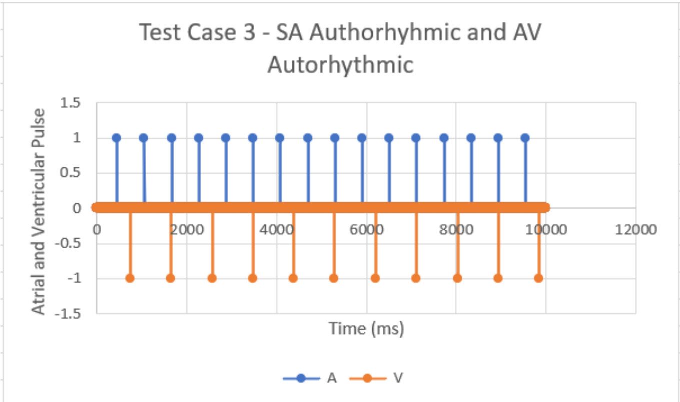

Fig 4. Graph of Test Case 3 - SA Authorhyhmic and AV Autorhythmic

In figure 4, we expect negative pulses coming from the ventricular on the graph. This is due to a lack of natural ventricular events so it requires ventricular pacing from the pacemaker. The length of time between ventricular paces is also as expected as the URI determines the fastest rate the heart can operate. It doesn't matter if the natural atrial events are pulsing faster since any atrial events within the PVARP time will be ignored. Hence, the ventricular pace is not the same frequency as the natural atrial event.

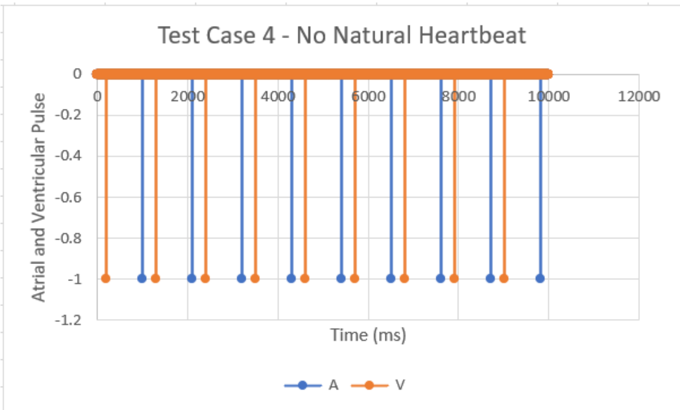

Fig 5. Graph of Test Case 4 - No Natural Heartbeat

In figure 5, we expect both the atrial and ventricular to be paced since there are no natural atrial or ventricular events.

General Difficulties Experienced

When trying to generate interrupts on the code for Mode 1, the buttons while being labelled 1 and 0 when actually pressing on them through the sample code the output on the DE2115 board the button displayed were 5 and 7. To work around this we had to expand the coverage of IRQ MASK.

On compilation of Nios in Quartus there was an issue where when compiling on a different system there would be issues as the Drive letter and user using this is not inclusive of the other teammates drive. To work around this we had to compile this project on a single user's account/computer and when working separately have the code transported via a text file.

Our SCChart files are located in ddd-mode-pacemaker-303-group-10\SCChart\DDD-mode/DDDpacemake2.sctx.

Our Nios code is located in ddd-mode-pacemaker-303-group-10\SCChart\DE2-115\software folder and the BSP code is located in ddd-mode-pacemaker-303-group-10\SCChart\DE2-115\HeLlOwOrLD_bsp folder.

Hours Dedicated

This is an estimate of the hours dedicated to each task from each member. There was equal contribution in this project.

- Planning: 5 hours
- SCCharts: 40 hours
- Nios - Mode 1: 12 hours
- Nios - Mode 2: 15 hours
- Documentation: 6 hours

Future work

A perceived avenue for future work would be to eliminate the need for polling when checking if the RX request is ready to receive data and instead use an interrupt.

Another perceived avenue is the functionality of being able to differentiate between a pacing or a natural beat.