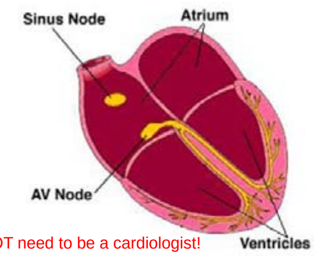
CS5219 Term Project
Pacemaker: Modeling and Verification

Lecturer: Abhik Roychoudhury
TA: Lei JU
National Univ. of Singapore

For more questions about the requirements document, contact julei@comp.nus.edu.sg

Human Heart



You do NOT need to be a cardiologist!

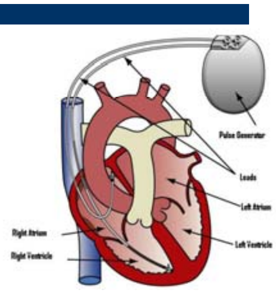
Natural Pacemaker

- An electric signal (pulse) is generated at the Sinus Node which stimulate left/right atrium to contract
- AV Node receives the signal, and after a delay (AV delay), it causes left/right ventricle to contract
- Automatically responds to body's changing need for oxygen

Bradycardia

- The beats per minute (bpm) rate of the heart is under the expectation (e.g., 60 bpm)
- Natural pacemaker is malfunctioning
  - No pulse (signal) is generated by Sinus node
  - Or signal is not strong enough to stimulate the AV node
- Medical solution: artificial pacemaker

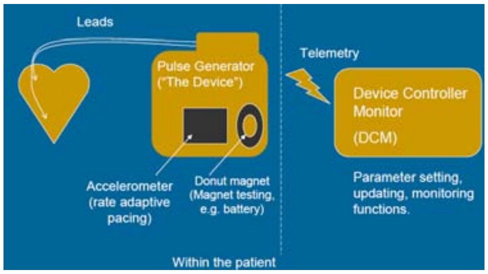
Pacemaker



Pacemaker

- Medical device which uses electrical impulses to fix abnormal heart rate
- Implanted in to patient body, configured according to the patient needs
  - Maintain adequate heart rate
  - Monitor patient's activity, dynamically change the heart rate requirement

Pacemaker Components



* By J. Fitzgerald @VSRnet Workshop at ABZ 2008

Pacemaker Components

- Device:
  - Pulse generator controller: maintain AV synchrony
  - batteries
- Device Controller-Monitor (DCM)
- Leads: wires that both sense and discharge electric pulses.
- Accelerometer: unit inside the device measuring body motion.

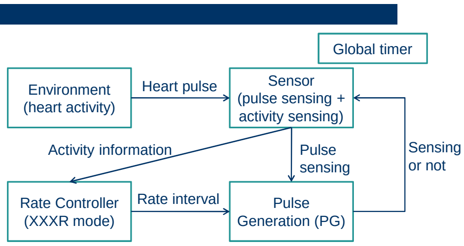
Pacemaker Formal Methods Challenge

- Hosted by SQRL (Software Quality Research Laboratory)
  - http://sqrl.mcmaster.ca/pacemaker.htm
- Based on a released informal specification of a previous generation of pacemaker by Boston Scientific
  - Spec.: http://sqrl.mcmaster.ca/pacemaker_spec.htm
  - Wiki: http://www.cas.mcmaster.ca/wiki/index.php/Pacemaker
- Deliverables: from formal specification to complete pacemaker software running on specified hardware

Goal of the Term Project

- Model (part of) the pulse generation controller with formal specification language (Promela)
- Use model checker (Spin) to verify the correctness and desired properties of your model

Model Overview [3]



Architecture

- Sequential model:

  timer->heart->sensor->rate controller(optional)

  ->PG->timer
- Concurrent model:

  timer || heart || sensor || rate controller || PG

  (Concurrent processes are considered to execute interleaving with nondeterministic)

  Synchronization and atomic actions may be needed

Bradycardia Operating Modes

	I	II	III	IV(optional)
Category	Chambers Paced	Chambers Sensed	Response to Sensing	Rate Modulation
Letters	O – None A – Atrium V – Ventricle D – Dual	O – None A – Atrium V – Ventricle D – Dual	O – None T – Triggered I – Inhibited D – Tracked	R – Rate Modulation

- Total 18 available working modes
  - DDDR,VDDR, DDIR, DOOR, VOOR, AOOR, VVIR, AAIR, DDD, VDD, DDI, DOO, VOO, AOO, VVI, AAI, VVT and AAT

Response to Sensing

- No Response To Sensing (O)
  - Pacing without sensing is asynchronous pacing. During asynchronous pacing, paces shall be delivered without regard to senses
- Triggered Response To Sensing (T)
  - During triggered pacing, a sense in a chamber shall trigger an immediate pace in that chamber.

Response to Sensing

- Inhibited Response To Sensing (I)
  - During inhibited pacing, a sense in a chamber shall inhibit a pending pace in that chamber.
- Tracked Response To Sensing (D)
  - During tracked pacing, an atrial sense shall cause a tracked ventricular pace after a programmed AV delay, unless a ventricular sense was detected beforehand.

Programmable Parameters

- LRL (Lower Rate Limit) - number of pace pulses delivered per minute in the absence of sensed activity in an interval starting at a paced event.
  - LRL interval: Longest safe interval (in ms) between two consecutive paces: 60000/LRL
- URL (Upper Rate Limit)
  - URL interval is the minimum time between a ventricular event and the next ventricular pace.

Programmable Parameters

- AV Delay: the shortest period from an atrial event to a ventricular pace.
- Atrial Refractory Period: for single chamber atrial modes, this is the time interval following an atrial event during which time atrial events shall not inhibit or trigger pacing

Programmable Parameters

- Please refer to [1] for all programming parameters and relevant operating modes
  - Read [2] and [3] for examples
  - You can use the nominal values (page 34 [1])
- You do NOT need to model all parameters listed in [1]
  - Depend on which modes you will model
  - If you fail to model any of the (necessary) parameters of a mode, try to discuss the reasons

Model Global Timer

```
Timer++;        /*increase timer by 1 millisecond*/
/*you may want to maintain a timer for a specific
    programmable parameter*/
if
::AVD_Timer >= 0 -> AVD_Timer ++; /*activate*/
::else -> ; /*inactivate*/
fi;
```

* You may use the same (or different) AVD_Timer in Heart and PG components, depending on your model

Model Heart Activity

```
 if
:: heart == nondeterministic ->
  if
     ::(Timer – lastPace == NR) -> lastPace = Timer; AVD_Timer = 0;
     if
        :: pulseA ! 1;
        :: /*do nothing*/
     fi;
     :: (AVD_Timer == AVD) ->
      if
        :: pulseV ! 1; AVD_Timer = -1; /*inactivate AVD_timer once a V pulse occurs*/
        :: /*do nothing*/
     fi;
:: heart == dead -> …
:: heart == missV -> …
fi;
```

Incomplete model

Model Sensor

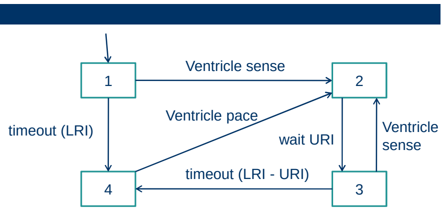
```
/*ASensor*/
if
   :: (mode == XAXX || mode == XDXX) ->
      do
      :: pulseA ? p  -> senseA ! 1;
      od;
    :: else -> ;
fi;
```

- Similarly for the VSensor

Example: VVI Mode

- V: Pace the ventricles
- V: Sense the ventricles
- I: During inhibited pacing, a sense in a chamber shall inhibit a pending pace in that chamber.

Example: VVI Mode [3]



LRI = 60000 / lower rate limit : maximum interval between two paces
URI = 60000 / upper rate limit : minimum interval between two paces

timout() in Spin

- No concept of “real-time” in Promela
- In Promela, timeout is a predefined global read-only variable, that has the value true in all global system states where no statement is executable in any active process, and false in all other states
- Carry no parameter. ~~timeout~~(5)

Safety Properties

- Pacemaker is a safety-critical system
- Design your own safety-properties and verify them on your model
  - No deadlock (Spin automatically checks deadlock)
  - Lower and upper rate limits
  - Verify correctness of modeling the programmable parameters
    - E.g., V pulse always occurs AVD ms after an A pulse --> [](AVD_Timer <= AVD)

Safety Properties

- The cardiologist decides the operating mode based on patient's symptom
  - You may have different requirements for different mode
- Some modes do not work in certain heart condition (environment)
  - E.g., AAT mode may not work for dead heart (no pulse generated)
- It is possible to use the verification results to guide the mode selection

Term Project

- Assessment (total 30 marks)
  - 15 marks for results and final report
  - 8 marks for innovative techniques,
  - 5 marks for final presentation,
  - 2 marks for interim report.

Minimum Expectations - 1

- 1 person project
  - Reasonable modeling and verification effort using SPIN checker.
  - Any abstractions used in modeling must be clarified properly.

Minimum Expectations - 2

- 2 person project
  - 1 person project expectations +
  - Using the SPIN model as a guide to generate C code.
  - You will also then be able to argue how/why you could use the SPIN model as guidance and the links between requirements, model and code.

Minimum Expectations - 3

- 3 person project
  - 2 person project expectations +
  - one group member writing the code without going through modeling and the two codes will be compared systematically to clarify coding errors or errors in understanding the requirements.

Project Deliverables

- One single .zip file includes
  - A report in .doc or .pdf format
  - Your pacemaker model:
    - .pml file(s) contain the your Promela model (necessary comments will be a plus)
    - LTL files contain properties to be verified

Project Deliverables: the Model

- Sequential model
- Consider the following heart behaviors as environment
  - Nondeterministic
  - Dead heart (no pulse signal generated)
  - Missing V pulse (signal is not strong enough to stimulate the AV node)
- Design your own safety properties (in separated LTL files) and verify them

Project Deliverables: the Model

- You should model at least SIX operating modes, including VOO, VVI, DDD, and AAT
- Model the XXXR mode (e.g., DDDR) earns you bonus credits ([1] Section 5.7)
- Model the Hysteresis pacing in any relevant modes earns you bonus credits ([1] Section 5.8)

Project Deliverables: Report

- Group members and Matric #
- Present all modes you have modeled
  - Explanation in text and/or state diagram (see [2] and [3] for examples)
- A table lists all programmable parameters you have modeled in your Promela specification
  - For each parameter, list in which mode(s) it is used

Project Deliverables: Report

- Summarize the critical properties you have designed to verify (see [3] for examples)
  - It is also interesting to discuss if any of the properties fails to hold
- Anything you want to explain/discuss about your modeling/experiences
- No more than 20 pages, single column

Conclusion

- Interesting project: real-world problem
- Difficulties:
  - unfamiliar application domain
  - Informal, incomplete, or even contradictory specification
  - Hint: self-learning/research, read the supplementary references

External References

[1] Pacemaker informal specification

[2] H Macedo, *Validating and Understanding Boston Scientific PACEMAKER Requirements*, Technical Report, Minho University, 2007

[3] L. A. Tuan, M. C. Zheng, and Q. T. Tho, *Modeling and Verification of Safety Critical Systems: A Case Study on Pacemaker*, SSIRI, 2010

[4] A Gomes and M Oliveira, *Formal specification of a cardiac pacing system*, FM, 2009

[5] H Macedo, P Larsen, and J Fitzgerald, *Incremental development of a distributed real-time model of a cardiac pacing system using vdm*, FM, 2008