Partnering to Provide Electronic Solutions for Full Life

Medtronic Microelectronics Center

Reliability Requirements for Implantable Medical Electronics

Mark Porter

28 October, 2009

Outline

- Market Characteristics
- Safety Standards
- Research Needs / Discussion Points
- Resilient System Characteristics
- Summary

Implantable Medical Device market characteristics continue to evolve

- **Reliability considerations are paramount**
  - Drive hardware / firmware design decisions
  - Heavily regulated industry (FDA, TUV, MHW, etc.)
  - Liability and litigation
- **Electronic content growth characteristics are driven by**
  - New therapy delivery options
  - Advanced DSP algorithms for therapy decision-making
  - Rich input data streams from new sensor technologies
  - Large volumes of diagnostic data for clinician review
- **Technology adoption lags the leading edge**
  - Implantable device companies are risk-averse
  - Although computing throughput and memory density are increasing, they are still distant from high-performance computing requirements
- **Ultra-low power dictates many constraints**

Internal standards have evolved over time to generate device reliability requirements and guarantee performance

- **Risk-based design decisions are made throughout the device hierarchy**
  - Patient impact (therapy delivery) is the top element
  - Device longevity is generally the second level
  - Comfort, clinician convenience make up the next level
- **IEC 61508 is being reviewed now as a way to standardize our processes across divisions**
  - Many of the processes spelled out in the standard are already in use across Medtronic
    - Risk/Hazard Analysis per system function
    - Safety Requirements
    - Reliability Allocation
    - Etc.
- **Lack of standardized reliability requirements for electronic components in life critical applications complicates supplier deliverables**
  - Use condition
  - Test method and qualification requirement

Our approach to design for reliability is also evolving

- **Transient logical faults are typically handled by Power On Reset (POR)**
  - System defaults to safe operating parameters
- **Memory systems have multiple levels of redundancy depending on criticality**
  - Device parameter settings and firmware opcode locations have robust ECC schemes
  - Patient and device data logging use block-level CRC
  - Scratch pad areas are generally not redundant
- **Backup performance for microcontroller failure reverts to 60 beats per minute**
  - Simple clock circuit continues to provide therapy
- **These approaches have been developed over time**
  - We are in the process of implementing a more robust system-level approach to Design for Reliability and Manufacturability (DRM)

Are interests appear very similar to automotive and aerospace in some areas

- **New, low power resilient architectures / methods**
  - Low power is a key differentiator
    - Specialized commercial semiconductor processes are getting closer to our requirements
    - Many defect types (latent and time-zero) that impact Medtronic do not present themselves to other Foundry customers
  - Low power defect detection / system degradation monitoring
  - Apply to “mature” technologies to achieve very high reliability levels
  - Mixed-signal and sensor architectures need to be folded into this equation
- **Firmware resiliency**
  - Firmware (faults) are part of the problem
  - Firmware (may be) part of the solution
- **Provable failure probability of fault-tolerant designs**
  - By design?
  - By demonstration?

Thanks to Tony Reipold, Kevin Kemp, and Claude Moughanni at Freescale for providing the impetus for these slides

Implantable ICs also have unique application condition, design constraints, and safety requirements

- **Failure Modes, Effects and Criticality Analysis**
  - What are the common failure modes in implantable IC?
    - Leakage?
    - Oxide crack?
    - EOS?
  - What are the most critical failure modes?
- **Manufacturing, storage, and use parameters that could affect reliability**
  - Mechanical
  - Thermal
  - Electrical
  - Environmental
  - Biocompatibility
- **Qualification**
  - Method (do we cover all possible failure mechanisms?)
  - Effectiveness (do we accelerate all possible failure mechanisms?)

What would a resilient implantable medical device system look like?

- **Current drain monitored in a known configuration**
  - Periodically stored in memory
  - Increase in current drain beyond some threshold kicks off system diagnostics
    - Determine if a faulty component exists
    - Re-route function to remove faulty component
    - Record information in memory
    - Notify doctor/patient that a device check is advised
  - Increase in battery depletion rate beyond known performance curves results in the same type of system diagnostic routine

What would a resilient implantable medical device system look like (continued)?

- **Periodic function check performed**
  - Within bounds of programmed parameters (i.e. do not need to check functions that are disabled)
  - Memory scan, op-code integrity, passive component check, RF module, lead impedance, high-energy capacitor check, battery voltage, etc.
  - Diagnostic data stored to memory
  - Degradation analysis performed
    - Critical timing, drive current, capacitance/ESR, etc.
  - Serious performance delta stored to memory
  - Notify doctor/patient that a device check is advised

What would a resilient implantable medical device system look like (continued)?

- **Software performance monitor to avoid firmware-induced Power On Resets (POR)**
  - – Software health monitor?
  - – Redundant application code
  - – Firmware located in multiple configurations
    - ▪ ROM vs. Flash vs. SRAM
  - – Hardware interrupt on bad instruction or memory location
    - ▪ Roll-back stack to known-good configuration
    - ▪ Maintain programmed parameters, rather than reset
  - – What types of software fault checking exist in current state of the art real-time operating systems?
  - – Fault diagnoses recorded to memory
  - – Notify doctor/patient that a device check is advised
  - – Re-load entire operating system in doctor’s office?

We have summarized the research needs of safety-critical applications into a number of common threads

1. **Develop a comprehensive library of fault-tolerance/resiliency techniques and map them onto architectural levels.**
2. **Build an error classification scheme that allows the physical, mechanistic view of failures to be categorized into types of errors. The resulting schema should mate directly with the library of design mitigation techniques in such a way that fault-tolerant architectures can be constructed.**
3. **Extend the visibility of errors/defects outside CMOS technology to encompass sensors, discrete components, passives, etc. A resilient design in safety-critical systems must have visibility into defects at all components, not just CMOS.**

Summary, continued

4. Perform research into fall-back techniques classified by application area. Safety-critical systems encompass very diverse areas such as implantable medical devices, automotive control, medical diagnostic equipment, nuclear plant control, etc. Not all techniques will be possible to implement in each of these industries.

5. Ensure research topics and solutions can be integrated with existing safety standards (e.g. IEC 61508). Although this may not be a specific area of focus, system designers will be increasingly required to establish the final safety level of their products with respect to a recognized standard. Understanding how any design strategy conforms to the standard will be critical in the future.

6. As system complexity increases with each design generation, the likelihood of unforeseen interactions among components also increases

Summary, continued

7. Build a system-level test and communication protocol that can transmit errors to the appropriate architectural level for disposition. Included in this area is the ability to understand how much of the hardware-level resiliency is being consumed at a given point in time, and how that fraction of available redundancy is changing over time (e.g. number of ECC bits, fraction of sub-block TMR errors, etc.). The protocol must be implementable on existing standards, or provide a mechanism to extend those standards to encompass new data content transmission (e.g. IEEE 1149.1).

8. Establish a cost-effective test vehicle for exploring resilient/fault-tolerant design techniques