Embedded App Properties and WCET

Static Properties of Commercial Embedded Real-Time Programs, (and Their Implication for WCET Analysis)

Jakob Engblom
Industrial PhD Student
Uppsala University / IT / Dept. of Computer Systems & IAR Systems AB (Uppsala, Sweden)
jakob@dona uu ee

Background

- How are embedded systems coded?
- How does this affect WCET analysis?

Properties of Embedded Programs

- Embedded programs contains:
  - Small, unsigned, global/static variables
  - Little dynamic data allocation
  - Mainly unsigned operations (bit-fiddling)
  - Many side-effect-only functions
  - Many simple functions
  - Use of embedded compiler features

Consequences for WCET Analysis

- Bad news
  - Recursion
  - Unstructured code
  - Function pointers
  - Data pointers
  - Complex loops
  - Complex decisions
  - Global variables
  - Non-termination

- Good news
  - Small data values
  - Simple & trivial functions
  - Simple loops
  - Simple decisions

Studied Programs

- 13 applications, 337 kloc
- Various industrial applications:
  - Telecomm, Vehicles, Consumer Products, ...
- Embedded, partially real-time programs
- Medium-capacity 8- and 16-bit CPUs:
  - Z80, 68HC11, C166, MELPS7000, H8, ...
- Medium-to-large European companies

Variables: Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>22.00%</td>
</tr>
<tr>
<td>Pointer</td>
<td>33.85%</td>
</tr>
<tr>
<td>Array</td>
<td>11.98%</td>
</tr>
<tr>
<td>Struct/Union</td>
<td>19.88%</td>
</tr>
<tr>
<td>Char</td>
<td>11.20%</td>
</tr>
<tr>
<td>Code Pointer</td>
<td>11.20%</td>
</tr>
<tr>
<td>Float</td>
<td>0.00%</td>
</tr>
</tbody>
</table>
**Variables: Size of Integers**

- Data size depends on CPU size
- Mostly bytes and shorts
- (Very different from desktop programs)

![Bar chart showing size of integers](chart1.png)

**Variables: Scopes**

By count:
- Parameters 29%
- Global 28%
- Auto 38%
- Static 5%

By bytes:
- Global 87%
- Static 2%
- Auto 7%
- Parameters 4%

![Pie charts showing variables](chart2.png)

**Operations & Variables: Sign and Size**

- Integer variables
  - Signed: 30%
  - Unsigned: 70%
- Integer operations
  - Comparators: 38%
  - Logic: 42%
  - Arithmetic: 20%

![Diagrams showing integer operations](chart3.png)

**Operations: Categories**

- Categories:
  - Compares: 49%
  - Arithmetic: 27%
  - Logic: 24%
- Ignored:
  - Jumps/Branches
  - Addressing
  - Loads/Stores

![Pie charts showing operations](chart4.png)

**Global Properties**

- About Embedded Programs
  - 0/13 use malloc & free
  - 5/13 use OS dynamic memory allocation
  - 9/13 use function pointers
  - 11/13 use intrinsic functions
  - 13/13 use the standard library

![Charts showing global properties](chart5.png)

**Functions: Unstructured Flow Graphs**

- Problem for certain compiler analyses and optimizations
- Can be caused by compiler
- 18 instances found in 5579 functions
  - Machine-generated code with gotos
  - Destructured by optimizer (relevant for WCET analysis tools)

![Diagrams showing flow graphs](chart6.png)
5579 functions in sample
14 loops (18 functions) found
- 4 user-interface code
- 10 machine generated protocol handlers
- Small CPUs, but with decent stacks
  - Z-80, 68HC11

Functions: Complexity

Functions: Recursive

Functions: Parameters & Returns

Loops: Exits

Consequences for WCET Analysis
**Future Work**

- Examine more real-life programs
- Examine benchmark suites:
  - EEMBC (EDN Embedded Microprocessor Benchmark Consortium)
  - UTDSP Suite
  - A few other leads to pursue

**More Information**

- Tech Report on MARE
- Paper at LCTES ’99:
  - Comparing Embedded to SpecInt95
- My homepage:
  - www.docs.uu.se/~jakob
  - Will have these slides online!

**Methodology**

- Modified IAR C compiler
- Replace code generator
- Mimic other compilers:
  - Size of int & pointers
  - Keywords (interrupt, near)
  - Intrinsics (di, ldx)
- Measuring optimized intermediate code

**Switches: Density vs Actual Cases**

- Actual cases: # of case X: labels
- All switches < 110 cases
- Little point in sophisticated code generation

**THE END!**

And then I use Excel to handle the data and draw graphs, conclusions.
Background

- Research: Worst-Case Execution Time analysis for real programs
- Idea: Quantitative investigation of real programs
- Result: the MARE project
  - Measurements of Actual Real-Time and Embedded Programs

Measurement Selection

- Static statistics
- No dynamic effects
- Each source file in isolation
- Measurement selection:
  - Most from WCET complications
  - Some from compilation
  - But... all of general interest