Class Review and Sample Exam Questions

CSE 237A

Prof. Tajana Simunic Rosing
Class Overview

• Plan for today:
  – Class review and sample questions from previous exams

• Upcoming:
  – HW3 due today
  – Exam the last day of class; no book/notes
    • Bring one 8 ½ x 11” sheet of paper with handwritten notes
  – Course evaluations are out!!!!
    • Please provide your feedback re. course – we take your feedback very seriously and look forward to hearing from you!
Embedded System Design

Hardware components

Concept

Specification

Software Components

HW/SW Partitioning

Estimation - Exploration

Design (Synthesis, Layout, ...)

Design (Compilation, ...)

Hardware

Software

Verification and Validation
Course Objectives

• Develop an understanding of the technologies behind the embedded computing systems
  – technology capabilities and limitations of the hardware, software components
  – methods to evaluate design tradeoffs between different technology choices.
  – design methodologies

• Overview of a few exciting research topics in embedded systems
• **Embedded development platforms**
  - ARM, RPi, Android, Arduino
• **CPUs, GPUs, DSPs**
• **Memory**
  - Caching, scratch pad, ARM mem. hierarchy, NVMs
• **Interfacing w peripherals**
  - Pooling, interrupts, DMA, GPIO, serial, I2C
• **AD/DA conversion**
  - Nyquist theorem, aliasing, quantization
• **Sensors, actuators (e.g. motors, servos, stepper motors, PWM)**
• **Timing & real-time scheduling**
  - Clock synchronization, logical clocks
  - Independent processes: EDF, RM etc.; Dependent processes: ASAP, ALAP, List scheduler
  - Priority inversion and inheritance
• **Real-time operating systems & Middleware**
  - VxWorks, FreeRTOS, RT-Linux; PALOS, TinyOS; uCOS-II, eCOS
• **Real-time IO**
  - Profibus, CAN, ARINC, TTP/A & C, FlexRay; wireless
• **HW/SW codesign**
• **Models of Computation**
  - StateCharts, SDL, PetriNets, data flow, SDFs, Esterel, Verilog/VHDL, UML
Logical clocks

• What are the scalar and vector time representation of point x?
Exam Review: Petri Net

• Initial marking: \([1 \ 0 \ 0 \ 0]\)
StateChart Problem

[StateChart Diagram]

- States: AWAKE, SLEEP, DISTURB, LOCAL, SYNC
- Transitions:
  - s': step
  - w
  - h': sleep
  - h'
  - h / disturbed
  - c
  - o
SDF

[Diagram of a network with nodes A, B, and C, and labeled edges with numbers 1, 2, 4, 8, a, b, c, and d.]
Resources and SDF

- Find the fastest and lowest energy schedule assuming SDF tasks have to be executed sequentially, on following HW:

<table>
<thead>
<tr>
<th>Task</th>
<th>Sensor node</th>
<th>DSP</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task A</td>
<td>1</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>Task B</td>
<td>-</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Task C</td>
<td>-</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

**Execution times (seconds)**

**Power consumption**

- **Static power**: 1W, 3W, 5W
- **Dynamic power**: 1W, 1W, 20W
Real-time Schedulers

<table>
<thead>
<tr>
<th>Task</th>
<th>Worst case execution time (WCET)</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>x</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>time</th>
<th>task</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>time</th>
<th>task</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dependent Task Scheduling

- Add/Sub = 1 cycle  Mul/Div = 2 cycles

\[ k = a+b+c+d \]
\[ l = e*f \]
\[ m = g+h+i*j \]
\[ x = k/n \]
\[ y = m*l \]
Esterel to FSM

Module Test
Input A, B, C, D, F, H;
Output E, G, O;

abort
  loop
    await B || await C
    present D emit E else abort
    await F; emit G;
    when H
  end
when A do
  emit O;
end module;