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

Specification

Concept

Software Components

Hardware

HW/SW Partitioning

Design

(Synthesis, Layout, ...)

Estimation - Exploration

Design

(Compilation, ...)

Verification and Validation

Software
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
Topics covered

• 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 0]
StateChart Problem
SDF

A \rightarrow B \rightarrow C

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>2</td>
<td>-4</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>6</td>
<td>-4</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>-3</td>
<td>2</td>
</tr>
<tr>
<td>d</td>
<td>6</td>
<td>0</td>
<td>-8</td>
</tr>
<tr>
<td>e</td>
<td>-3</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
```

\[ A = 2B \]
\[ 3B = 2C \]
\[ 3A = 4B \]
\[ 3B = 4C \]
\[ 4C = 6B \]

AAAABBCCCD
\[ \phi \ \phi \ 6 \ \phi \ 12 \] initial state
\[ [8 \ 12 \ 6 \ 24 \ 12] \] min. full size

9
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></th>
<th>Sensor node</th>
<th>DSP</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution times (seconds)</td>
<td></td>
<td></td>
<td></td>
</tr>
<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>
<tr>
<td>Power consumption</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static power</td>
<td>1W</td>
<td>3W</td>
<td>5W</td>
</tr>
<tr>
<td>Dynamic power</td>
<td>1W</td>
<td>1W</td>
<td>20W</td>
</tr>
</tbody>
</table>

\[
\text{min } T
\]

\[
\begin{array}{cccc}
\text{S} & 4.2 + 4.1 + 24.1 & \text{D} & 4.3 + 4.4 + 4.24K \\
\text{C} & 4.5 + 4.5 + 12.25C & \text{D} & 4.3 + 4.4 + 4.24K \\
\end{array}
\]
# Real-time Schedulers

<table>
<thead>
<tr>
<th>Task</th>
<th>WCET</th>
<th>Period=Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>unknown</td>
</tr>
</tbody>
</table>

Min period of C for RM schedule = 8

<table>
<thead>
<tr>
<th>Time</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>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>Task</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>
Dependent Task Scheduling

- Add/Sub = 1 cycle  Mul = 2 cycles;
- Assume there are 2 multipliers, one adder
- Design a list schedule using mobility as a task priority metric

\[
\begin{align*}
O_1 &:= i_1 \times i_2 + i_3 \times i_4 \\
O_2 &:= O_1 + i_5 \\
O_3 &:= i_6 + i_7 \\
O_4 &:= O_3 \times i_8 \\
O_5 &:= O_2 + O_4
\end{align*}
\]
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;