**CODE GENERATION AND OPTIMIZATION**

Every Computer Architect should know how a compiler generates code to maximize the efficiency of the processor (micro) architecture. In CGO, you will learn the basics of an optimizing compiler and implement your own optimization passes in a production compiler (LLVM). The topics include control-flow analysis, data-flow analysis, and instruction selection and scheduling, register allocation and a variety of optimizations.

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Topic</th>
<th>Handouts</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aug 28th</td>
<td>Welcome</td>
<td>R1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sept 2nd</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Sept 4th</td>
<td>Lexical Analysis</td>
<td>H1</td>
<td>R1</td>
</tr>
<tr>
<td>3</td>
<td>Sept 9th</td>
<td>Parsing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sept 11th</td>
<td>Intermediate Representation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Sept 16th</td>
<td>Intro to Control Flow</td>
<td>H1(a)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Sept 18th</td>
<td>Loop Structures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Sept 23rd</td>
<td>Traces/Superblocks</td>
<td>H2</td>
<td>H1(b)</td>
</tr>
<tr>
<td>6</td>
<td>Sept 25th</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Sept 30th</td>
<td>If-conversion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Oct 2nd</td>
<td>Liveness</td>
<td>H3</td>
<td>H2</td>
</tr>
<tr>
<td>7</td>
<td>Oct 7th</td>
<td>Reaching Definitions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Oct 9th</td>
<td>Available Definitions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Oct 14th</td>
<td>Static Single Assignment</td>
<td>H4</td>
<td>H3</td>
</tr>
<tr>
<td>8</td>
<td>Oct 16th</td>
<td>SSA-Based Optimizations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Oct 21st</td>
<td>Top-Down/Bottom-Up</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Oct 23rd</td>
<td>Graph Coloring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Oct 28th</td>
<td><strong>Midterm I</strong></td>
<td>P1</td>
<td>H4</td>
</tr>
<tr>
<td>10</td>
<td>Oct 30th</td>
<td>Instruction Selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Nov 4th</td>
<td>Classic Optimizations</td>
<td>P1.a</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Nov 6th</td>
<td>Loop Optimizations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Nov 11th</td>
<td>Instruction Scheduling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Nov 13th</td>
<td>List/Global Scheduling</td>
<td>P1.b</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Nov 18th</td>
<td>Software Pipelining</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Nov 20th</td>
<td>I/D-Cache Optimizations</td>
<td>P1.c</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Nov 25th</td>
<td>DOALL Parallelism</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Nov 27th</td>
<td>DOACROSS Parallelism</td>
<td>P1.d</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Dec 2nd</td>
<td>Compiler-driven DVFS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Dec 4th</td>
<td><strong>Midterm II</strong></td>
<td>P1.e</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Dec 9th</td>
<td>Code Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Dec 11th</td>
<td><strong>Final Report</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Key: P(roject updates); H(omeworks); R(reading assignments – there will be more)

Instructor: Vijay Janapa Reddi (ENS 530), vj@ece.utexas.edu
Teaching Assistant: Jingwen Leng (ENS 623), jingwen@utexas.edu, Riddhi Shah (ENS 110), riddhi.j.shah@utexas.edu

Course Code: EE382V
Unique ID: 17285

Course Times: MW 1:30pm – 3:00pm, ENS 126
Office Hours: TBD
TA Hours: TBD

Grading Policy:
- Class Problems / Reading Assignments: 20%
- Programming Assignments: 20%
- Midterm 1: 20%
- Midterm 2: 20%
- Project: 20%

Pre-reqs: 460N (Computer Architecture)