RE-HASE: Regular-Expressions Hardware Synthesis Engine

Mohamed El-Hadedy*, Xinfei Guo**, Xiaoping Huang ^^, Martin Margala**

*Department of Computer Science, University of Virginia, Charlottesville, Virginia, USA
**Department of Electrical and Computer Engineering, University of Virginia, Virginia, USA
*Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA
**Department of Electrical and Computer Engineering, University of Massachusetts Lowell, MA, USA
^^Department of Electrical and Computer Engineering, Northwestern Polytechnic University, China

Email: hadedy@Illinois.edu, mea4c@virgina.edu
What is a Regular Expression?

• A regular expression (regex) describes a pattern to match multiple input strings.
• Regular expressions descend from a fundamental concept in Computer Science called finite automata theory
• Regular expressions are endemic to Unix
• Some utilities/programs that use them:
  – vi, ed, sed, and emacs
  – awk, tcl, perl and Python
  – grep, egrep, fgrep
  – compilers

• The simplest regular expression is a string of literal characters to match.
• The string matches the regular expression if it contains the substring.
What is a DFA and NFA?

An **NFA** is a **Nondeterministic Finite Automaton**. Nondeterministic means it can transition to, and be in, multiple states at once (i.e. for some given input). A **DFA** is a Deterministic Finite Automaton. Deterministic means that it can only be in, and transition to, one state at a time (i.e. for some given input).
Code Automatically Generation

- **State Transition**: Construct the state machine transition
- **State Condition**: Extract the matching condition between different DFA states and drive the state to jump
- **Match Report**: Collect the matching information and output onto the bus
FPGA internal Architecture: parallel matching
Regrouping for shared Memory Architecture

Benchmark (1)

Benchmark (2)
FPGA Synthesis Results for Benchmarks with Xilinx-V7 VC707 Board

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Critical Path Delay (ns)</th>
<th>Logic Latency (ns)</th>
<th>Route Latency (ns)</th>
<th>Utilization (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dotstar005</td>
<td>1.257</td>
<td>0.256</td>
<td>1.001</td>
<td>2%</td>
</tr>
<tr>
<td>dotstar010</td>
<td>1.269</td>
<td>0.26</td>
<td>1.009</td>
<td>2%</td>
</tr>
<tr>
<td>dotstar020</td>
<td>1.294</td>
<td>0.269</td>
<td>1.025</td>
<td>2%</td>
</tr>
<tr>
<td>Spyware_put</td>
<td>1.34</td>
<td>0.26</td>
<td>1.08</td>
<td>1.2%</td>
</tr>
<tr>
<td>backdoor</td>
<td>1.42</td>
<td>0.25</td>
<td>1.17</td>
<td>3.5%</td>
</tr>
<tr>
<td>Benchmark</td>
<td>Critical Path Delay (ns)</td>
<td>Static Power (mW)</td>
<td>Dynamic Power (W)</td>
<td>Total Power (W)</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Dotstar005</td>
<td>0.16</td>
<td>0.9582</td>
<td>0.284</td>
<td>0.2851</td>
</tr>
<tr>
<td>dotstar010</td>
<td>0.20</td>
<td>0.9612</td>
<td>0.285</td>
<td>0.2860</td>
</tr>
<tr>
<td>dotstar020</td>
<td>0.17</td>
<td>0.9513</td>
<td>0.282</td>
<td>0.2829</td>
</tr>
<tr>
<td>spyware_put</td>
<td>0.28</td>
<td>0.2807</td>
<td>0.082</td>
<td>0.0822</td>
</tr>
<tr>
<td>backdoor</td>
<td>0.24</td>
<td>0.1883</td>
<td>0.083</td>
<td>0.0831</td>
</tr>
</tbody>
</table>
Concullsion

- Python Tool (RE-HASE), which can transfer the NFA/DFA to RTL.
- Ability to direct the RTL to either ASIC/FPGA based on the application
- Novel features for better optimization regarding using resources without decreasing the total performance
Future Work

- Using the regrouping feature with the coherency interfaces such as CAPI (IBM), QPI (Intel) to improve the total performance of the RegEx Engine based on the RE-HASE
- Using the ability of producing the same feature for using small devices for the IoT applications
- Integrating RE-HASE to HLS applications as a library
Current research

- El-Hadedy, Mohamed; Guo, Xinfei; Hwu, Wen-Mei; Stan, Mircea; Skadron, Kevin. Crypt-Pi: A Light and Fast Crypto-Processor for IoT Applications, TECHCON-2017, Austin, TX, September 20-22, 2017. (Best Paper Award)
- El-Hadedy, Mohamed; Guo, Xinfei; Stan, Mircea; Skadron, Kevin; Hwu, Wen-Mei. R-NNPE: Reconfigurable Neural Network Processing Elements, TECHCON-2017, Austin, TX, September 20-22, 2017
- El-Hadedy, Mohamed; Mihajloska, Hristina; Gligoroski, Danilo; Kulkarni, Amit; Stroobandt, Dirk; Skadron, Kevin. A 16-bit Reconfigurable Encryption Processor for Pi-Cipher. 23rd Reconfigurable Architectures Workshop, Co-located IPDPS 2016. (Best Paper Award)
- Kevin Angstadt Jack Wadden Xiaoping Huangy Mohamed El-Hadedy, Westley Weimer Kevin Skadron RAPID: Accelerating Pattern Search Applications with Reconfigurable Hardware, TECHCON-2016, Austin, TX, September 11-14, 2016 (Best Paper Award)
- El-Hadedy, Mohamed; Guo, Xinfei; Margala, Martin; Stan, Mircea; Skadron, Kevin, Dual-Data Rate Transpose-Memory Architecture Improves the Performance, Power and Area of Signal-Processing Systems. Submitted to JSPS Journal. (Accepted)
- El-Hadedy, Mohamed; Kulkarni, Amit; Stroobandt, Dirk; Skadron, Kevin, Reco-Pi: A Reconfigurable Crypto-processor for Pi-Cipher, in Journal of Parallel and Distributed Computing, Special issue on Reconfigurable Computing Through the Looking Glass, 2016. (Accepted) (Journal, JPDC, science-direct)
- El-Hadedy, Mohamed. Runtime Flexibility in FPGAs (Opportunities and Challenges), PhD trial lecture, Trondheim, Norway, February 2012.
Thank you

www.recoiot.com