Module 3: Behavioral Synthesis
Lecture 3.1: Overview

Anand Raghunathan
raghunathan@purdue.edu

Fall 2014, ME 1052, T Th 12:00PM-1:15PM
Behavioral Synthesis: 10,000 ft view

- Input: Algorithm
- Output: Architecture (RTL)

Euclid’s GCD algorithm

GCD RTL circuit

Also called: Architectural Synthesis, High-level Synthesis, ESL Synthesis
Behavioral Synthesis: Historical Perspective

- Roots lie in the ideas proposed by Mario Barbacci and Daniel Siewiorek in the early 1970s
  - Idea: “Compile” an instruction set specification (in the ISPS language) into hardware

Mario Barbacci
Daniel Siewiorek
http://www.cs.cmu.edu/~dps/

CMU RT-CAD System

ISP description of 8-bit shift-add multiplier

Graph representation

RTL implementation
Behavioral Synthesis: Historical Perspective

• In the 1970’s and 1980’s Silicon compilation was the holy grail of EDA researchers – and industry
• Wave of academic / industrial research systems:
  – System Architect’s Workbench (CMU)
  – Cathedral I, II, III (IMEC / K. U. Leuven)
  – MIMOLA (TU Dortmund)
  – Yorktown Silicon Compiler (IBM)
  – Olympus (Stanford)
  – ...
• Early 1990s: Textbooks on Behavioral Synthesis
  – Gajski et al., Camposano et al.
• Mid-1990’s saw the first generation of commercial behavioral synthesis tools
Behavioral Synthesis: Historical Perspective

- The ideas and tools were oversold and/or ahead of their time
  - No burning need
  - Technologies not “universal” (restricted to a small class of circuits, e.g., DSP)
- 2005: Synopsys shuts down Behavioral Compiler product

“Whither (or Wither) HLS?”
“High-level synthesis: A retrospective”, Rajesh Gupta and Forrest Brewer, Chapter 2 in High-level Synthesis book by Coussy and Morawiecz
Behavioral Synthesis: Historical Perspective

• A new beginning
  – The emergence of SoCs and System-level design methodology has created a new need
  – New languages, algorithms, faster computers (and expectations) have led to a revival of behavioral synthesis
Behavioral Synthesis: Input

- Behavioral description
  - No notion of time (or loosely timed)
  - No structure

```
ARCHITECTURE algorithm of barcode IS
BEGIN
barcode: PROCESS
BEGIN
  eoc <= false; memw <= false;
  data <= 0; addr <= 0;
  RESET_LOOP : LOOP
      LOOP EXIT WHEN start; END LOOP;
      MAIN_LOOP: LOOP
          LOOP EXIT WHEN scan; END LOOP;
          flag := wh; actnum := 0;
          white := 0; black := 0;
          eoc <= false;
          LOOP
              IF pixel = wh THEN
                  white := white +1;
                  IF flag = bl THEN
                    actnum := actnum+1;
                    memw <= false;
                  ELSE
                    memw <= true;
                  END IF;
              ELSE
                  black := black + 1;
                  IF flag = wh THEN
                    actnum := actnum + 1;
                    memw <= false;
                  ELSE
                    memw <= true;
                  END IF;
                  white := 0; flag := bl;
                  data <= black;
              END IF;
          END LOOP;
          addr <= actnum;
      EXIT WHEN (white = 255) OR (black = 255); END LOOP MAIN_LOOP;
      EXIT WHEN (actnum = num) AND (white = 255); END LOOP MAIN_LOOP;
      memw <= false; eoc <= true;
      EXIT WHEN start = false; END LOOP;
  END LOOP RESET_LOOP;
END PROCESS;
END ALGORITHM;
```
Behavioral Synthesis: Output

- Structural RTL that can be synthesized using logic synthesis tools
Behavioral Synthesis

• Why?
  – Reduced design complexity
  – Better architecture exploration
  – Bridging abstraction gap between HW and SW enables tradeoffs and co-design
Behavioral Synthesis vs. RTL Coding

- Automatically select “architecture” and bypass RTL coding / verification / debugging loop
Reduced Design Complexity

- Example: Configurable/Extensible processor (NEC)
  - Much faster design time and simulation speed

<table>
<thead>
<tr>
<th>Configurable processor</th>
<th>C(HLS)</th>
<th>RTL(Man.)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>#lines</td>
<td>1.3KL</td>
<td>9.2KL</td>
<td>7.6X</td>
</tr>
<tr>
<td>Sim. speed</td>
<td>61Kc/s</td>
<td>0.3K cycles/sec</td>
<td>203X</td>
</tr>
<tr>
<td>Size</td>
<td>19KG</td>
<td>18KG</td>
<td>5%+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Viterbi decoder</th>
<th>C(HLS)</th>
<th>RTL(Man.)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>5 KG</td>
<td>50 KG</td>
<td>1/10</td>
</tr>
</tbody>
</table>

Source: K. Wakabayashi, NEC
Behavioral Synthesis: Better Architecture Exploration

- Scope of optimization beyond what can be achieved at the logic (gate) or transistor level

Basic design choice in HLS – performance vs. area/power by regulating parallelism

```
char A,B,C,D;
char E,F;
main(){
    char X;
    X = A + B;
    E = X * D;
    F = (B + C) * X;
}
```

8 Lines

Logic synthesis

1 cycle
Delay: 2T

RTL

Constraint 1

+ : 2
* : 2

Constraint 2

+ : 1
* : 1

100 lines

3 cycle
Delay: 1T

Courtesy: K. Wakabayashi, NEC
Behavioral Synthesis: Better Architecture Exploration

RTL: Same architecture
Number of Registers, Adders, Buses, etc

Beh. Synth.: Different architectures
Steps in Behavioral Synthesis

• Refinement in two dimensions
  – From Untimed to Timed: ______________________
  – From Functional to Structural: ________________