SYNTHESIS from PRODUCTION-BASED SPECIFICATIONS*

Andrew Seawright and Forrest Brewer
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106

*Supported in part by the California Micro Program #90-195 and Synopsys® Inc.
BASIC IDEA

Specification of a Design Using Hierarchy of Productions
Each Production is Sub-Machine

Behavior Due to:

1. Composition of the Sub-Machines
2. HDL Clauses Attached to the Production Grammar

Hardware Analog of Popular Software Techniques
The Production-Based Specification Compiled to VHDL
EXAMPLE

port { ...interface information...  }

process_front {
    wait until clock'event and clock = '1';
    if  (xc = '1' and xd = '0') then PBS_TOKEN := A;
    elsif (xc = '1' and xd = '1') then PBS_TOKEN := B;
    elsif (xc = '0' and xd = '1') then PBS_TOKEN := C;
    elsif (xc = '0' and xd = '0') then PBS_TOKEN := D;
    end if;
}

...additional stuff...

:::

    mouse    ->  .* event;
    event    ->  forward | reverse;
    forward  ->  A B+ C+ D;  { x <= x + 1;  }
    reverse  ->  D C+ B+ A;  { x <= x - 1;  }

:::
EXAMPLE

Compiled Machine
RELATED WORK

Software Tools: Yacc and Lex

M. A. Jackson

Ullman et. al.

Devadas and Keutzer
METHODOLOGY and ADVANTAGES

Productions form Natural Partitioning of Design Behavior

Concise Specification of Protocol Engines, Controllers

Ensemble Behavior Determined by Additive Facets of Behavior

Descriptive Partitioning of Design
SYSTEM

User

Production Based Specification

Tool

PBS Compiler

Procedural VHDL

Simulate

Synthesis

Structural VHDL

Simulate
MODEL

interface

\{tokens\}

\begin{align*}
\text{"Data Path"} & \quad \text{"Controller"} \\
synchronous
\end{align*}
BEHAVIOR MODEL

HDL Actions Viewed by Designer As:
Combinationally Executed in Single Clock Cycle
Executing at the Designated Points in the Protocol

Primitive Actions Conceptually Execute Before Abstract

Ex: 
\begin{align*}
\text{block} & \rightarrow \text{byte}^4; \quad \{ y := 0; \} \\
\text{byte} & \rightarrow \text{bit}^8; \quad \{ y := y + 1; \}
\end{align*}

TRANSFORMATIONS!

Any Transformation OK if Behavior Same
HARDWARE vs. SOFTWARE

Timing and Performance Constraints

Lookahead

Specification of Continuous Behavior

Exceptions
EXCEPTION OPERATORS

Ex: \[ p \rightarrow a!b; \]
While in p, if events which can’t be described by production a or any other production, then production b active.

Ex: \[ p \rightarrow a!!; \]

Exception Operators:

Provide Access to Productions’ Complement Space
Are Resolved when Deterministic Controller Constructed
EXCEPTION SCOPING

M → ... A!B ...;
A → ... C!D ...;

“death”
PBS COMPILATION

PBS
  ↓
Collapse Productions
  ↓
Intermediate DAG
  ↓
Thompson’s Construction
  ↓
NFA

NFA
  ↓
Subset Construction
  ↓
DFA
  ↓
Resolve Exceptions
  ↓
FSM Optimization
  ↓
VHDL
library work;
use work.<name>_pak.all;

entity <name> is
port
);

architecture BEHAVIOR of <name> is
architecture_decl{}

begin
PBS_MACHINE: process
declarations...
decl{}
begin
process_front{}

machine core...

process_end{}
end process;
end BEHAVIOR;

trailer{}}
ADD BEHAVIORS...

::

mouse -> .* event;

event -> forward | reverse | pause;

pause -> A A | B B | C C | D D;
   { idle_time <= idle_time + 1; }
forward -> A B+ C+ D; { x <= x + 1; idle_time <= 0; }
reverse -> D C+ B+ A; { x <= x - 1; idle_time <= 0; }

::
SYNTHESIS

(mouse2 / area minimized)
## EXPERIMENTS

<table>
<thead>
<tr>
<th>metric</th>
<th>mouse1</th>
<th>mouse2</th>
<th>cache</th>
<th>parity</th>
<th>bounce</th>
<th>count0</th>
<th>pager2</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Productions</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>17</td>
<td>5</td>
<td>5</td>
<td>21</td>
</tr>
<tr>
<td>No. Actions</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>39</td>
</tr>
<tr>
<td>lines of productions and actions</td>
<td>4</td>
<td>5</td>
<td>11</td>
<td>21</td>
<td>9</td>
<td>4</td>
<td>139</td>
</tr>
<tr>
<td>PBS size (lines)</td>
<td>38</td>
<td>45</td>
<td>41</td>
<td>48</td>
<td>36</td>
<td>41</td>
<td>187</td>
</tr>
<tr>
<td>procedural VHDL (lines)</td>
<td>117</td>
<td>142</td>
<td>83</td>
<td>120</td>
<td>96</td>
<td>108</td>
<td>1269</td>
</tr>
<tr>
<td>No. NFA states</td>
<td>25</td>
<td>37</td>
<td>18</td>
<td>1020</td>
<td>13</td>
<td>30</td>
<td>1688</td>
</tr>
<tr>
<td>No. DFA states</td>
<td>7</td>
<td>9</td>
<td>3</td>
<td>16</td>
<td>5</td>
<td>4</td>
<td>536</td>
</tr>
<tr>
<td>Transitions with actions</td>
<td>2</td>
<td>10</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>548</td>
</tr>
<tr>
<td>CPU (Sec.)</td>
<td>0.1*</td>
<td>0.2*</td>
<td>0.1*</td>
<td>2.0*</td>
<td>0.1*</td>
<td>0.1*</td>
<td>18.9*</td>
</tr>
<tr>
<td>Standard Cells</td>
<td>62</td>
<td>115</td>
<td>9</td>
<td>44</td>
<td>13</td>
<td>29</td>
<td>**</td>
</tr>
<tr>
<td>Relative Area</td>
<td>188</td>
<td>313</td>
<td>23</td>
<td>99</td>
<td>42</td>
<td>79</td>
<td>**</td>
</tr>
</tbody>
</table>
CONCLUSIONS
Production-Based Specification and Synthesis
Model and Implementation Presented

FUTURE WORK
Optimization of Data Flows
High Level Synthesis
Productions of Multiple Token Streams
Current Research

Remove the Abstraction of Interface in the current Token Specification Method

Target Interacting Machines

Utilize the Production Hierarchy in Structuring the Machines