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;
}
```

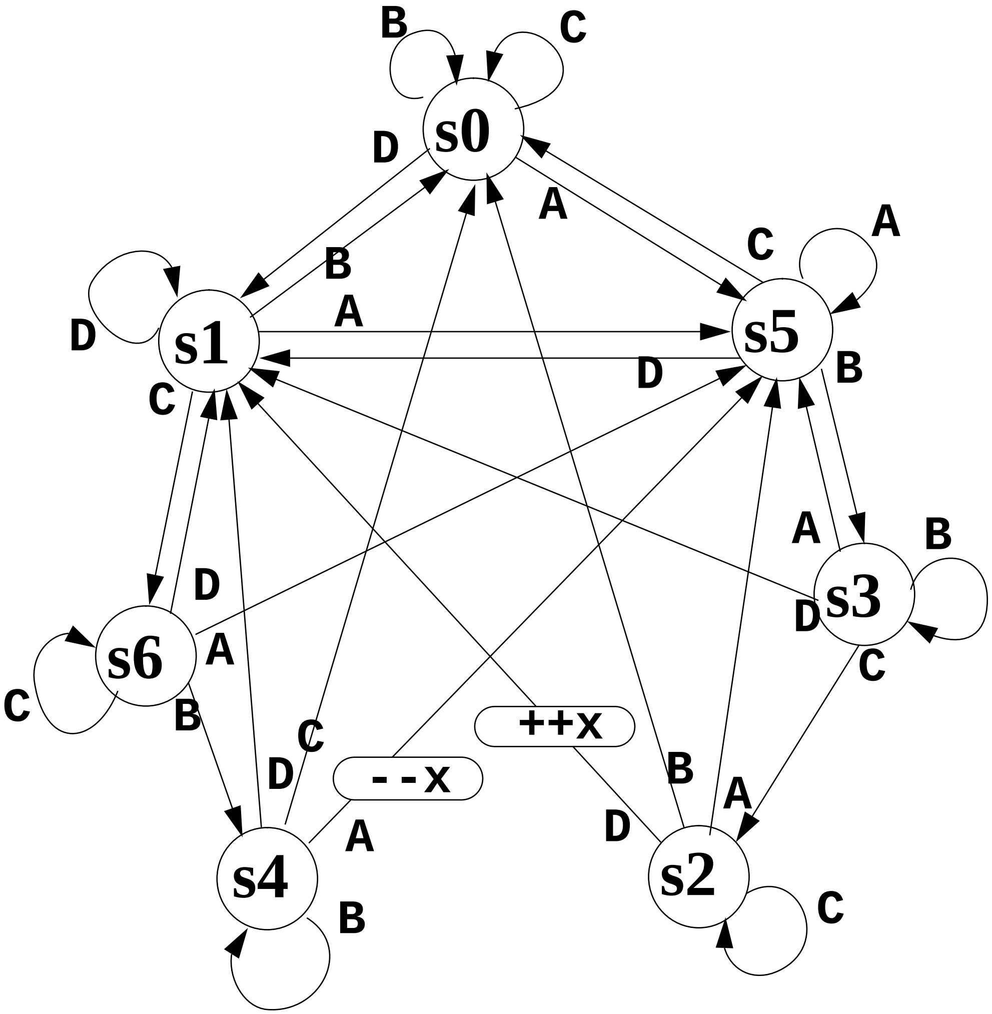
...additional stuff...

```
::

            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

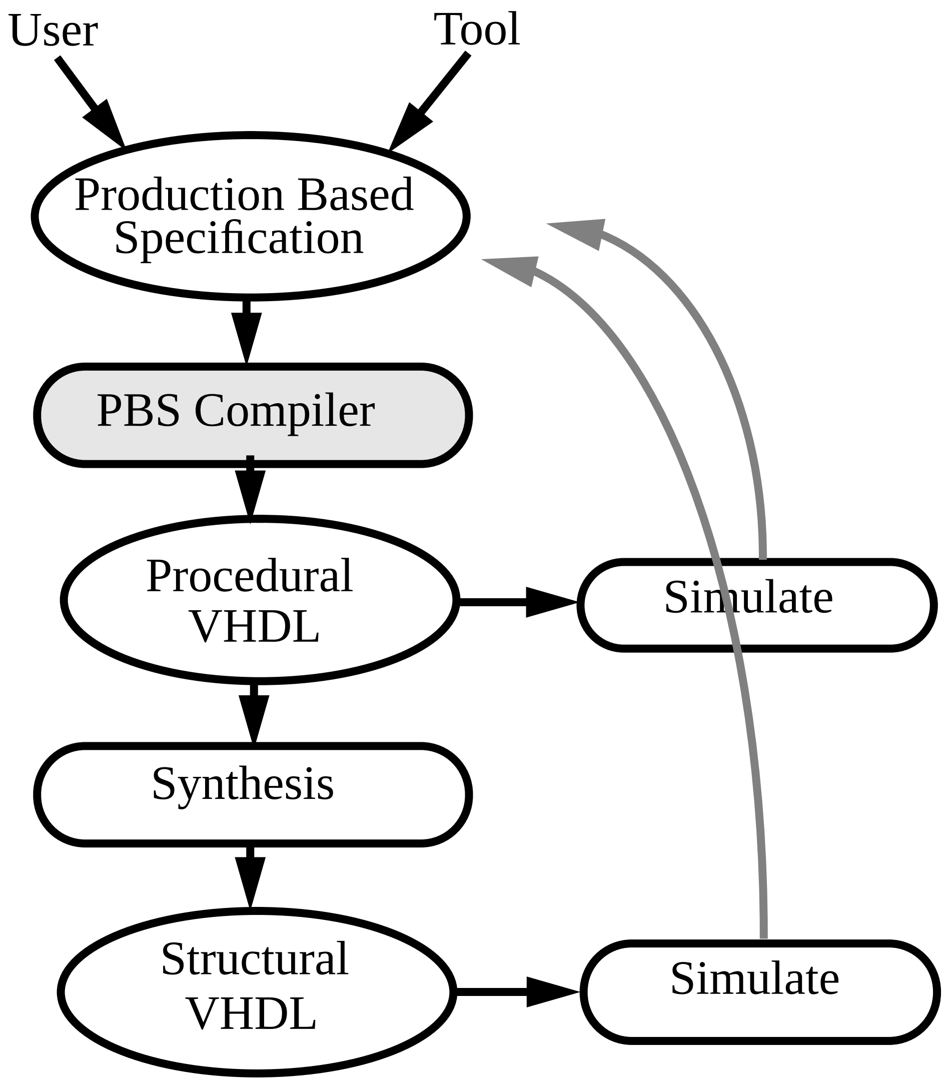
Productions form Natural Partitioning of Design Behavior

Concise Specification of Protocol Engines, Controllers

Ensemble Behavior Determined by Additive Facets of Behavior

Descriptive Partitioning of Design

SYSTEM



User

Tool

Production Based Specification

PBS Compiler

Procedural VHDL

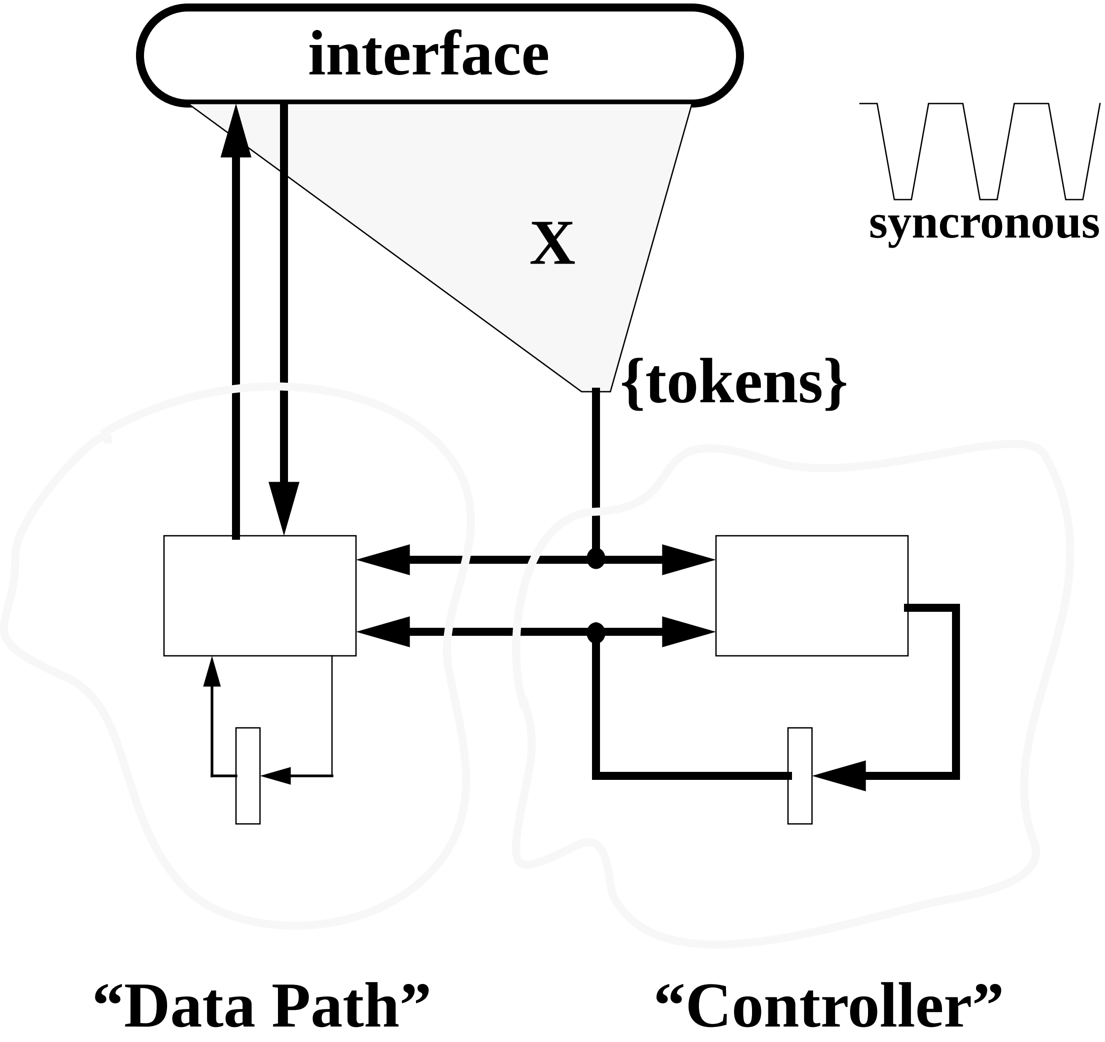
Simulate

Synthesis

Structural VHDL

Simulate

MODEL



interface

X

syncronous

{tokens}

“Data Path”

“Controller”

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:

```
block -> byte^4;  { y := 0; }
byte  -> bit^8;   { y := y + 1; }
```

TRANSFORMATIONS!

Any Transformation OK if Behavior Same

HARDWARE vs. SOFTWARE

Timing and Performance Constraints

Lookahead

Specification of Continuous Behavior

Exceptions

EXCEPTION OPERATORS

Ex: `p -> 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 -> a!!;`

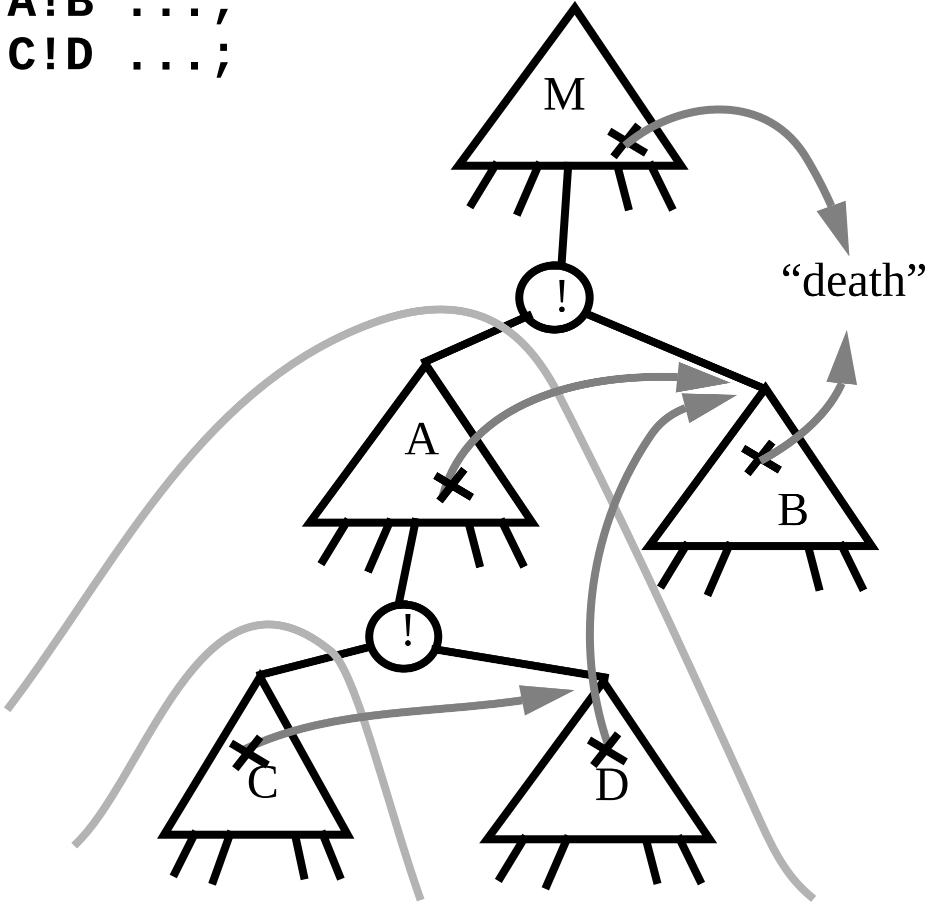
Exception Operators:

Provide Access to Productions’ Complement Space

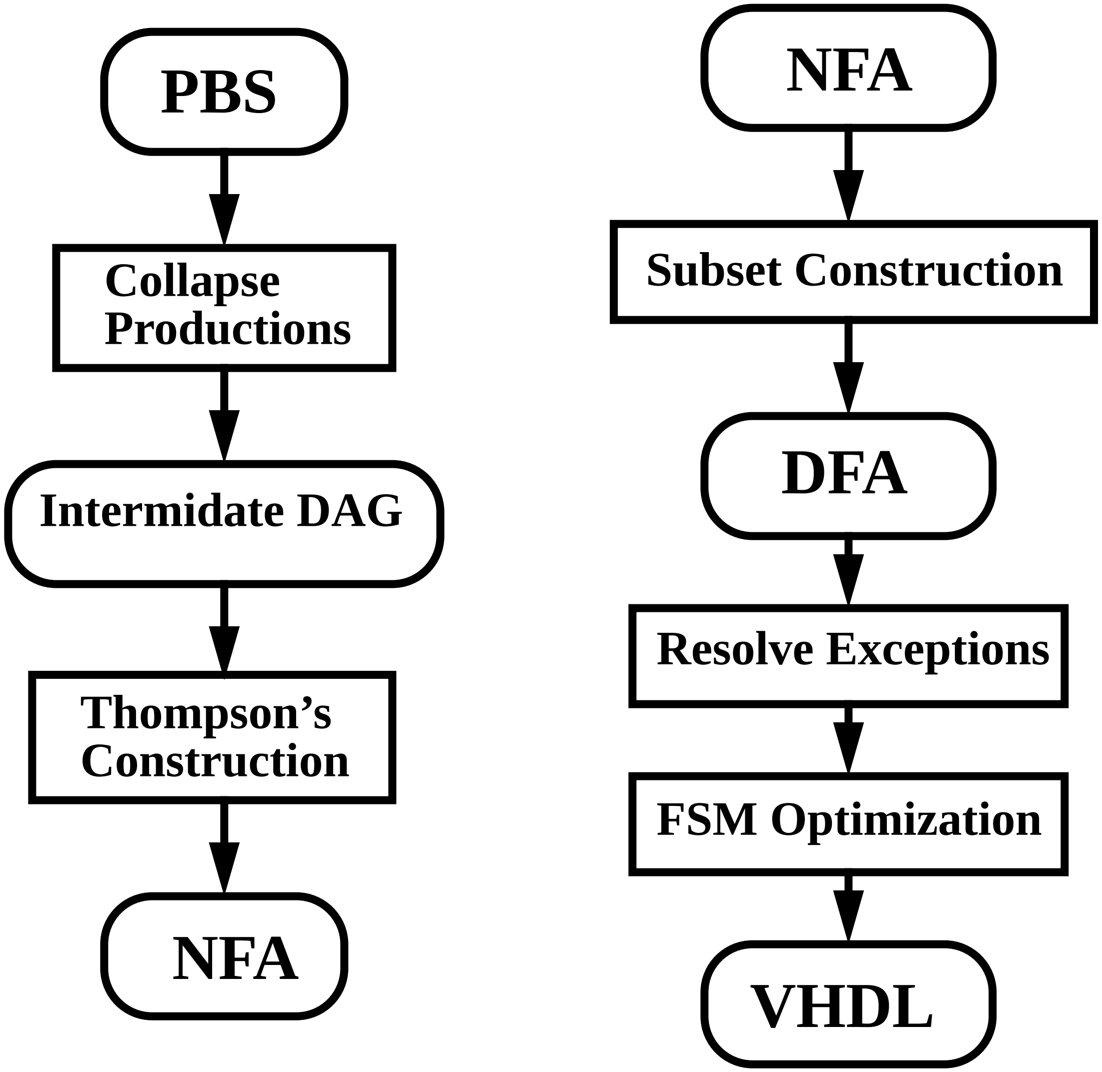
Are Resolved when Deterministic Controller Constructed

EXCEPTION SCOPING

```
M -> ... A!B ...;
A -> ... C!D ...;
```



PBS COMPILATION



PBS

Collapse Productions

Intermidate DAG

Thompson's Construction

NFA

NFA

Subset Construction

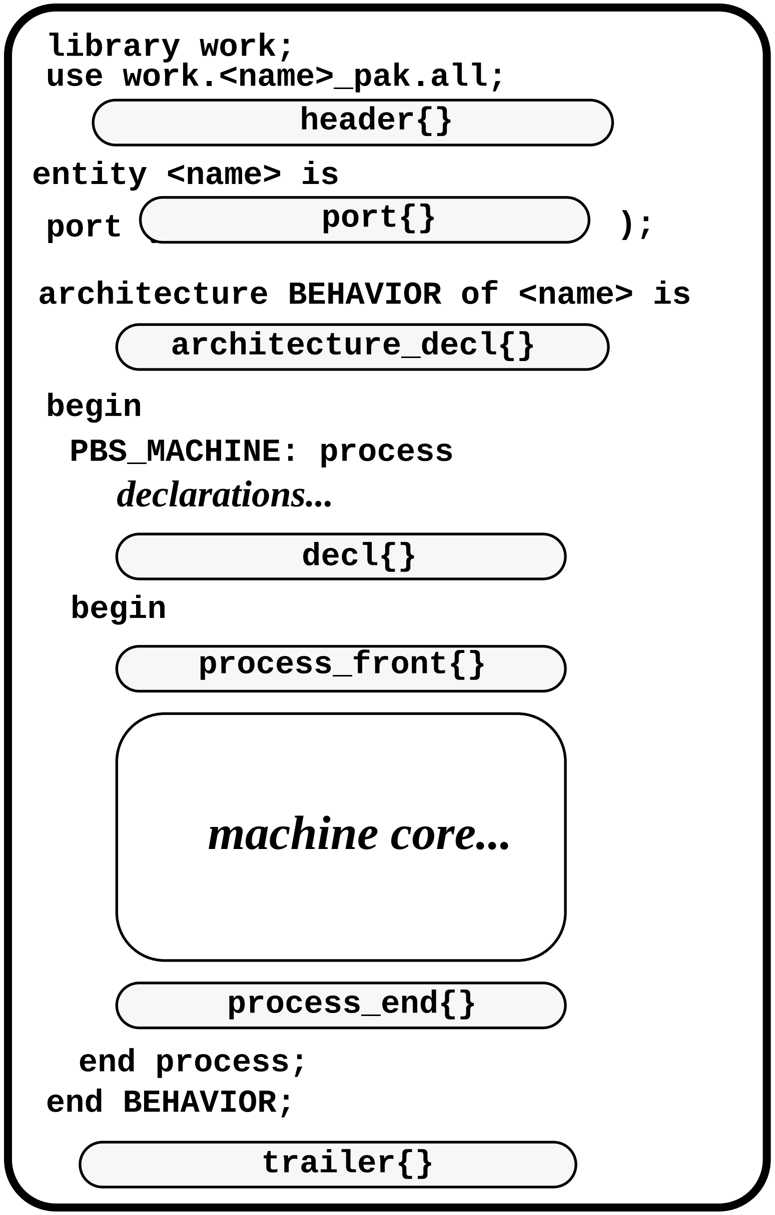
DFA

Resolve Exceptions

FSM Optimization

VHDL

VHDL SKELETON



```
library work;
use work.<name>_pak.all;
     header{}
entity <name> is
port (    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

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

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