The Synthesis of Combined Mealy and Moore Machines Structural Model Using Values of Output Variables as Codes of States

Adam Klimowicz, Valery Salauyou
Bialystok University of Technology
Faculty of Computer Science
Bialystok, Poland
Agenda

1. Introduction
2. Basic models of finite state machines
3. Proposed approach
4. Algorithm
5. Example
6. Experimental results
7. Conclusions
Introduction

In practice, the designers of sequential circuits the most widely use two types of finite state machines:

- Mealy automaton
- Moore automaton

The model of FSM, where vectors of output signals are the same as codes of internal states is also known (Medvedev, Glushkov).

An application of registers on inputs and outputs of machine for buffering of input signals and synchronization of output signals with clock was also described (Pomeranz, Cheng).
Basic structural models of finite state machines

Every vector of output signals \( w_t \) of Moore FSM is the same as a code of its internal state \( a_t \).
Proposed approach (1)

- All the transitions from the state $a_i$, are formed by the same vector of values of output variables $w_i$ (a code or part of the code of state $a_i$).
- Then in the transition of a machine from the state $a_i$ in the register RG will be set a code of $a_i$, while at the outputs of register RG will be generated an output set $w_i$.
- This fact allows to use the outputs of memory elements that match a single vector of values $w_i$ as a FSM outputs.
- Requirements:
  - a configuration of output macrocells of PLD, which implement their elements of memory in a register output mode
  - in the same macrocell of PLD will be realized an output variable and a feedback variable.
Proposed approach (2)

Using the output vector $w_i$ as a code (or part of the code) of state $a_i$ requires the following prerequisites:

1) Set of $w_i$ should not be formed on the other transitions of the state machine, except from transitions from the state $a_i$.

2) All output variables, taking values ‘1’ in the vector $w_i$, should be output variables of the Moore machine, so they should not depend on the input variables.

- A violation of the first condition can lead to an unexpected move of the state machine into a state $a_i$ (a violation of determination of behaviour of a FSM).
- A violation of the second condition makes it impossible to change the values of output variables of the Mealy machine when the values of input variables are changing, then it leads to disruption of the functioning of the state machine.
Coincidence of timing diagrams in AC model of FSM

- Let in the beginning of the machine cycle be a state machine transition to stable internal state, when on the outputs of memory elements valid values are set.

- In traditional implementation of Mealy finite state machines on PLD at the start of cycle output signals are formed not earlier than the delay time $t_{PD}$ caused by combinational part of PLD.

- In the proposed approach, the outputs of a Moore automaton defined by the internal state code, will be formed at the beginning of machine cycles, i.e. at least for the time $t_{PD}$ faster than the output signals of Mealy FSM.

- In most cases of the practical use of finite state machine that forming of the output signals is valid.
The method of synthesis – assumptions (1)

- \( A \) – a set of all internal states of a FSM
- \( A_B \) – a set of Moore FSM internal states
- \( A_A \) – a set of Moore FSM internal states, \( A_A = A \setminus A_B \)
- \( y(a_i) \) – a set of output variables, where ‘1’ values are formed on the transitions from the state \( a_i \)
- \( Y_A \) – a set of output variables, which values equal '1' are formed on the transitions from the states of Mealy-type machine from the set \( A_A \)
  \[
  Y_A = \{ y(a_i) | a_i \in A_A \}
  \]
- \( Y_B \) – a set of output variables of Moore machine, \( Y_B = Y \setminus Y_A \).
Assumptions (2)

- The output variables of the set $Y_A$ are directly dependent on the input variables and are asynchronous with the input variables, i.e. the values of the input variables may change the values of output variables of Mealy machine.
  - Realization on combinational outputs of PLD

- The output variables of the set of Moore FSM $Y_B$ do not directly depend on input variables and their values change with the change of internal state of a finite state machine. The output variables of a Moore machine can act as a code or a part of code of internal states.
  - Realization on register outputs of PLD
The algorithm of synthesis

1) Create the set of output variables $Y_B$ of Moore machine. If $|Y_B|=0$, go to step 6.

2) Construct a matrix $W$. The rows of $W$ correspond to the states of a FSM, the columns – to output variables of the set of Moore FSM $Y_B$. For the states of the set $A_B$ in the corresponding rows of the matrix $W$ the output values of the set of Moore FSM $Y_B$ are stored. For the states of the Mealy FSM (set $A_A$) in the corresponding rows of the matrix $W$ undefined values should be written (dashes).

3) Solve the problem of extension of rows of the matrix $W$ with binary code of minimal length $R$ so that all rows of the matrix $W$ are mutually orthogonal.

4) The codes of the internal states of a finite state machine should be determined. To do this, each state $a_i$ should be associated with the code $K(a_i)$ equal to the value of row $i$ of the matrix $W$.

5) Construct logical equations of transition and output functions of a FSM.

6) The end.
Example (1)

\[ y(a_1) = \{y_1\}, \]
\[ Y_A = \{y_1\} \]
\[ Y_B = Y \setminus Y_A, \quad \rightarrow \quad Y_B = \{y_2, y_3\}. \]

Variables \( y_2 \) and \( y_3 \) are Moore FSM variables.

Their values can be used as a part of the code of a state from \( A_B \) set.
Example (2)

Matrix W

<table>
<thead>
<tr>
<th></th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_2$</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>$a_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$a_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$a_3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$a_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$a_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$K(a_1) = (-1)$;

$K(a_2) = (110)$;

$K(a_3) = (100)$;

$K(a_4) = (010)$;

$K(a_5) = (000)$.

Realization in programmable logic

Traditional method – 6 MCs:
- 3 MCs for output functions, 3 MCs for transition functions, 46 gates

AC Model – 3 MCs, 24 gates
### Example (3)

#### Structural transition table

<table>
<thead>
<tr>
<th>$a_m$</th>
<th>$K(a_m)$</th>
<th>$X(a_m,a_s)$</th>
<th>$a_s$</th>
<th>$K(a_s)$</th>
<th>$Y(a_m,a_s)$</th>
<th>$Y(a_m)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1e_2e_3$</td>
<td>$d_1d_2d_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_1$</td>
<td>- - 1</td>
<td>$x_1$</td>
<td>$a_2$</td>
<td>1 1 0</td>
<td>$y_1$</td>
<td>-</td>
</tr>
<tr>
<td>$a_1$</td>
<td>- - 1</td>
<td>$\bar{x}_1$</td>
<td>$a_3$</td>
<td>1 0 0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$a_2$</td>
<td>1 1 0</td>
<td>$x_2$</td>
<td>$a_4$</td>
<td>0 1 0</td>
<td>-</td>
<td>$y_2, y_3$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>1 1 0</td>
<td>$\bar{x}_2$</td>
<td>$a_2$</td>
<td>1 1 0</td>
<td>-</td>
<td>$y_2, y_3$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>1 0 0</td>
<td>$x_3$</td>
<td>$a_5$</td>
<td>0 0 0</td>
<td>-</td>
<td>$y_2$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>1 0 0</td>
<td>$\bar{x}_3$</td>
<td>$a_4$</td>
<td>0 1 0</td>
<td>-</td>
<td>$y_2$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>0 1 0</td>
<td>1</td>
<td>$a_5$</td>
<td>0 0 0</td>
<td>-</td>
<td>$y_3$</td>
</tr>
<tr>
<td>$a_5$</td>
<td>0 0 0</td>
<td>1</td>
<td>$a_1$</td>
<td>- - 1</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Logic equations

- $y_1 = e_3 \cdot x_1$;
- $d_1 = e_3 + e_1 \cdot e_2 \cdot e_3 \cdot x_2$;
- $d_2 = e_3 \cdot x_1 + e_1 \cdot e_2 \cdot e_3 + e_1 \cdot e_2 \cdot e_3 \cdot x_3$;
- $d_3 = e_1 \cdot e_2 \cdot e_3$. 
Experiments (1)

MAX 3000 CPLD family
C1, D1 – a number of LC and max. delay for traditional method
C2, D2 – a number of LC and max. delay for method which uses AC model

<table>
<thead>
<tr>
<th>FSM</th>
<th>C1</th>
<th>C2</th>
<th>C1/C2</th>
<th>D1</th>
<th>D2</th>
<th>D1/D2</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyb</td>
<td>26</td>
<td>20</td>
<td>1,30</td>
<td>10,8</td>
<td>14,5</td>
<td>0,74</td>
</tr>
<tr>
<td>lion9</td>
<td>7</td>
<td>5</td>
<td>1,40</td>
<td>13,0</td>
<td>10,3</td>
<td>1,26</td>
</tr>
<tr>
<td>pma</td>
<td>40</td>
<td>21</td>
<td>1,90</td>
<td>21,6</td>
<td>11,4</td>
<td>1,89</td>
</tr>
<tr>
<td>s820</td>
<td>42</td>
<td>38</td>
<td>1,11</td>
<td>21,3</td>
<td>14,6</td>
<td>1,46</td>
</tr>
<tr>
<td>s832</td>
<td>49</td>
<td>42</td>
<td>1,17</td>
<td>21,3</td>
<td>13,1</td>
<td>1,63</td>
</tr>
<tr>
<td>shifreg</td>
<td>4</td>
<td>3</td>
<td>1,33</td>
<td>10,2</td>
<td>7,5</td>
<td>1,36</td>
</tr>
<tr>
<td>tma</td>
<td>24</td>
<td>14</td>
<td>1,71</td>
<td>15,7</td>
<td>10,3</td>
<td>1,52</td>
</tr>
<tr>
<td>average</td>
<td></td>
<td></td>
<td>1,42</td>
<td></td>
<td></td>
<td>1,41</td>
</tr>
</tbody>
</table>
Experiments (2)

FLEX 10KE FPGA family
C3, D3 – a number of LC and max. delay for traditional method
C4, D4 – a number of LC and max. delay for method which uses AC model

<table>
<thead>
<tr>
<th>FSM</th>
<th>C3</th>
<th>C4</th>
<th>C3/C4</th>
<th>D3</th>
<th>D4</th>
<th>D3/D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyb</td>
<td>102</td>
<td>95</td>
<td>1.07</td>
<td>38.4</td>
<td>43.3</td>
<td>0.89</td>
</tr>
<tr>
<td>lion9</td>
<td>23</td>
<td>20</td>
<td>1.15</td>
<td>21.9</td>
<td>19.6</td>
<td>1.12</td>
</tr>
<tr>
<td>pma</td>
<td>107</td>
<td>143</td>
<td>0.75</td>
<td>40.7</td>
<td>48.4</td>
<td>0.84</td>
</tr>
<tr>
<td>s820</td>
<td>255</td>
<td>145</td>
<td>1.76</td>
<td>53.1</td>
<td>40.9</td>
<td>1.30</td>
</tr>
<tr>
<td>s832</td>
<td>257</td>
<td>167</td>
<td>1.54</td>
<td>52.7</td>
<td>45.5</td>
<td>1.16</td>
</tr>
<tr>
<td>shifreg</td>
<td>10</td>
<td>3</td>
<td><strong>3.33</strong></td>
<td>20.1</td>
<td>12.3</td>
<td><strong>1.63</strong></td>
</tr>
<tr>
<td>tma</td>
<td>69</td>
<td>68</td>
<td>1.01</td>
<td>37.1</td>
<td>33.4</td>
<td>1.11</td>
</tr>
<tr>
<td><strong>average</strong></td>
<td></td>
<td></td>
<td>1.52</td>
<td></td>
<td></td>
<td>1.15</td>
</tr>
</tbody>
</table>
Conclusions

- A deviation from the terms of the absolute coincidence of the timing diagrams of output signals for the combined model of finite state machine can significantly expand the set of combined models that allow an efficient implementation on PLD.

- Future research:
  - a development of new structural models of the combined finite state machines and the methods of their synthesis on PLDs
  - a development of new structural models of finite state machines that allow to effectively use the features of modern PLDs with advanced architecture.