PLS

Low Level FSM Entries
# Table of contents

**Low Level FSM Entries**

1. PLS/FSM Synthesis
   1.1. PLS/FSM language
   1.2. Example : PLS/FSM Moore model
   1.3. Example : PLS/FSM Mealy model
   1.4. PLS/FSM synthesis using PLS

2. FSM / KISS - Berkeley Synthesis
   2.1. KISS Syntax : Mealy Controllers
   2.2. KISS syntax : Moore controllers
   2.3. Remarks on "Don't Care" States
   2.4. KISS -Berkeley synthesis using PLS
Low Level FSM Entries
1. PLS/FSM Synthesis

The PLS/FSM description language is used to define both Moore and Mealy controllers in a concise manner.

1.1. PLS/FSM language

The syntax of the PLS/FSM language is the following:

```
<PLS/FSM Controller> ::= GRAPH <Name> (MEALY / MOORE)
                        <Input Declaration>
                        <Output Declaration>
                        ( <State Declaration> ) +
                        END

<Input Declaration> ::= INORDER <Input Name> ("," <Input Name>) + ";"

<Output Declaration> ::= OUTORDER <Output Name> ("," <Output Name>) + ";"

<State Declaration> (for Moore Controller) ::= STATE <State Name> (":" <Command List>) ";"
                                             ( <Transition Declaration> ) +

<Transition Declaration> (for Moore Controller) ::= ( <Expression> / TRUE ) ";" <Next State Name> ";"

<Command List> (for Moore Controller) ::= ( ( DC / "?" ) <Output Name> (IF <Expression>))
                                         ("," ( DC / "?" ) <Output Name> (IF <Expression>)) *

<State Declaration> (for Mealy Controller) ::= STATE <State Name> ";"
                                             ( <Transition Declaration> ) +

<Transition Declaration> (for Mealy Controller) ::= ( <Expression> / TRUE ) ";" <Command List> ";" <Next State Name> ";"

<Command List> (for Mealy Controller) ::= ( ( DC / "?" ) <Output Name> )
                                         ("," ( DC / "?" ) <Output Name> ) *

<Expression> ::= Expressions in the PLS/FSM language are built up using ",", ";", "!" for the negation, "+" for the Or and "." for the And.
Example: ((a + b).c + !d).e is an expression
```

Figure 1: PLS/FSM language syntax
1.2. Example: PLS/FSM Moore model

```
GRAPH ex1 moore
INORDER test, d0, d1, d2 ;  # <Input Declaration>
OUTORDER c0, c1, c2 ;     # <Output Declaration>
STATE e0 : c0 ;           # Declaration of state e0,
                          # associated output is c0
  test : e1;              # transition from e0 to e1 if test = 1
  !test : e0;
STATE e1;                  # Declaration of state e0, no associated output
  !test : e1;
  test.!d0.!d1.!d2 : e3;
  test.!d2.!d1.d0 : e1;
STATE e3 : DC c1, c2 IF d0 ;  # Declaration of state e3,
                          # associated outputs are c1 and c2
                          # c1 is a Don’t Care output,
                          # c2 is emitted if d0 is 1 (conditional output)
  test : e3;
  !test : e0;
END
```

Figure 2: PLS/FSM Moore example

1.3. Example: PLS/FSM Mealy model

```
GRAPH cse mealy
INORDER p1, p2 ;            # <Input Declaration>
OUTORDER s1, s2, s3 ;      # <Output Declaration>
STATE st0 ;                # Declaration of state st0
  p1.p2 : : st0 ;           # transition from st0 to st0 if p1.p2 = 1
  # no output emitted
  p1.!p2 : s1,s3 : st2 ;   # transition from st0 to st2 if p1.!p2 = 1
  # associated output are s1 and s3
  !p1 : DC s1 : st1 ;      # transition from st0 to st1 if !p1 = 1
  # s1 is a Don’t Care output
STATE st1 ;
  p1 : : st0 ;
  !p1 : s2 : st1 ;
STATE st2 ;
  TRUE : s1,s2 : st0 ;     # unconditional transition from st2 to st0
END
```

Figure 3: PLS/FSM Mealy example
1.4. PLS/FSM synthesis using PLS

From the command menu of the Graphical User Interface (GUI), select the execute menu. Then select the "PLS_FSM" format from the list of available inputs (Input Format). Using the source option button, specify the desired encoding algorithm and flip-flop type parameters as shown in figure 4.

Then using the selection boxes, select all other appropriate synthesis parameters. Refer to the technology pages of the user manual for details on the technology specific options.

![Figure 4: Source option selection for PLS/FSM synthesis](image-url)
2. FSM / KISS - Berkeley Synthesis

The PLS extended KISS format allows to define both Moore and Mealy controllers. "Don’t care" can be defined for output vectors and current and next states.

2.1. KISS Syntax : Mealy Controllers

In a Mealy controller, outputs are associated with transitions. The syntax of a KISS file for Mealy controllers is defined as follows (comments start with a #) :

\[\text{.i <Input Number>} \]
\[\text{o <Output Number>} \]
\[\text{.ilb <Input Name List>} \]
\[\# \text{name of the inputs, separator is space, tab or CR} \]
\[\# \text{this list is optional ; if omitted, default names will be assumed} \]
\[\text{.ob <Output Name List>} \]
\[\# \text{name of the outputs, separator is space, tab or CR} \]
\[\# \text{this list is optional ; if omitted, default names will be assumed} \]
\[\text{.reset <Reset Name>: optional, if omitted MASTER_RESET will be assumed} \]
\[\text{.clock <Clock Name>: optional, if omitted MASTER_CLOCK will be assumed} \]
\[\# \text{the clock sense and the reset polarity will be selected during the synthesis process} \]
\[\text{.p <Transition number>} \]
\[\text{s <State Number>} \]
\[\text{.type <TypeValue>} \]
\[\# \text{optional section defining the meaning of the digits in the} \]
\[\# \text{<Next State> and <Output Vector> fields} \]
\[\# \text{For more information, see the Truth-Table pages.} \]
\[\# \text{Default value is assumed to be "fd"} \]
\[\{ \text{<Input Condition> <Current State> <Next State> <Output Vector>} \} * \]
\[\# \text{number of lines is <Transition number>} \]
\[\# \text{<Input Condition> : string with length <Input Number>} \]
\[\# \text{containing '0' '1' or '-'} \]
\[\# \text{<Current State> is either a symbolic name or * (Don't Care)} \]
\[\# \text{<Next State> is either a symbolic name or * (Don't Care)} \]
\[\# \text{<Output Vector> : string with length <Output Number>} \]
\[\# \text{containing '0' '1' or '-'} (\text{Don't Care}) \]
\[\text{.e} \]
\[\# \text{end} \]

Figure 5: KISS syntax for Mealy FSM

![Diagram of Mealy FSM example](image)

Figure 6: Mealy FSM example
2.2. KISS syntax : Moore controllers

In Moore controllers, outputs are associated with states and may depend on inputs (conditional outputs). The syntax of a KISS file for Moore controllers is defined as follows (comments start with a #):

```
.i <Input Number>
o <Output Number>
ilb <Input Name List>
ob <Output Name List>
.reset <Reset Name, optional, if omitted MASTER_RESET will be assumed>
clock <Clock Name, optional, if omitted MASTER_CLOCK will be assumed>
   # the clock sense and the reset polarity will be selected during the synthesis process
.p <Number of lines>
   # lines defining both transitions and output vectors
.s <State Number>        # same heading for Moore and Mealy controllers
   
   { <Input Condition> <Current State> <Next State> <Null Output Vector> } *
   # Definition of the transitions.
   # number of lines is <Transition Number>
   # <Input Condition> : string with length <Input Number>
   # containing '0', '1' or '-'.
   # <Current State> is either a symbolic name or * (Don't Care)
   # <Next State> is either a symbolic name or * (Don't Care)
   # <Null Output Vector> : string with length <Output Number>
   # containing only '0' (outputs are associated to states)

   { <Input Condition> <State> 'void' <Output Vector> } *
   # output definition
   # number of lines is <State Number>
   # <Input Condition> : string with length <Input Number>
   # containing '0' '1' or '-', this allows to define conditional outputs.
   # <Current State> is a symbolic name
   # 'void' indicates that we are defining the outout associated with
   # state <Current State> and therefore next state definition is useless.
   # <Output Vector> : string with length <Output Number>
   # containing only '0', '1' or '-' (Don't Care)
\e  # end
```

Figure 8: KISS syntax for Moore FSM
Low Level FSM Entries

2.3. Remarks on "Don't Care" States

KISS format "don't care" states are specified using the '*' character. A current-state "don't-care" condition indicates that, independent of the current state, a specified input produces a transition to specified next state and the specified output condition. "Don't-care" next states are usually specified if an input and current-state condition cannot occur. These outputs are indicated by a ".-".

Figure 11: Portion of KISS file with Don't care states

```
1000  *  RESET  0000
    # go to state RESET if bit 0 is true,
    # regardless of the current state
    # BEWARE: Note that this case corresponds to a
    # synchronous reset; asynchronous reset state will
    # be defined during the synthesis phase by using
    # the reset option of the encoding command
```
2.4. KISS -Berkeley synthesis using PLS

From the command menu of the Graphical User Interface (GUI), select the execute menu. Then select KISS-Berkeley format from the list of available inputs (Input Format). Using the source option button, specify the desired encoding algorithm and flip-flop type parameters as shown in figure 12.

Then using the selection boxes, select all other appropriate synthesis parameters. Refer to the technology pages of the user manual for details on the technology specific options.

![KISS Options](image)

Figure 12: Source option selection for Berkeley KISS synthesis