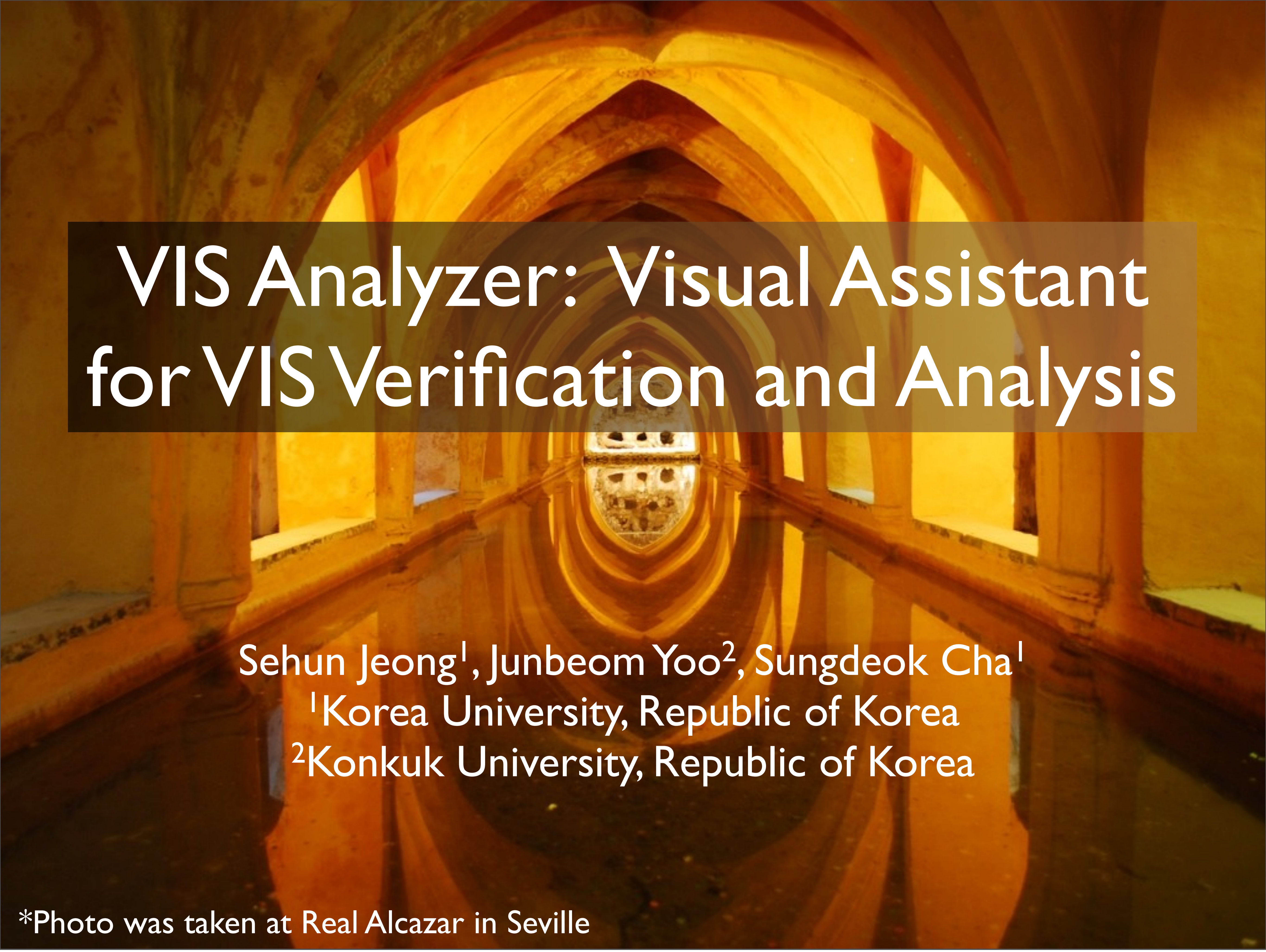


VIS Analyzer: Visual Assistant for VIS Verification and Analysis

Sehun Jeong[1], Junbeom Yoo[2], Sungdeok Cha[1]
[1]Korea University, Republic of Korea
[2]Konkuk University, Republic of Korea

*Photo was taken at Real Alcazar in Seville

Outline

- Motivation
- Our Solution
- Comparisons
- Conclusions & Future Work

KNICS Project

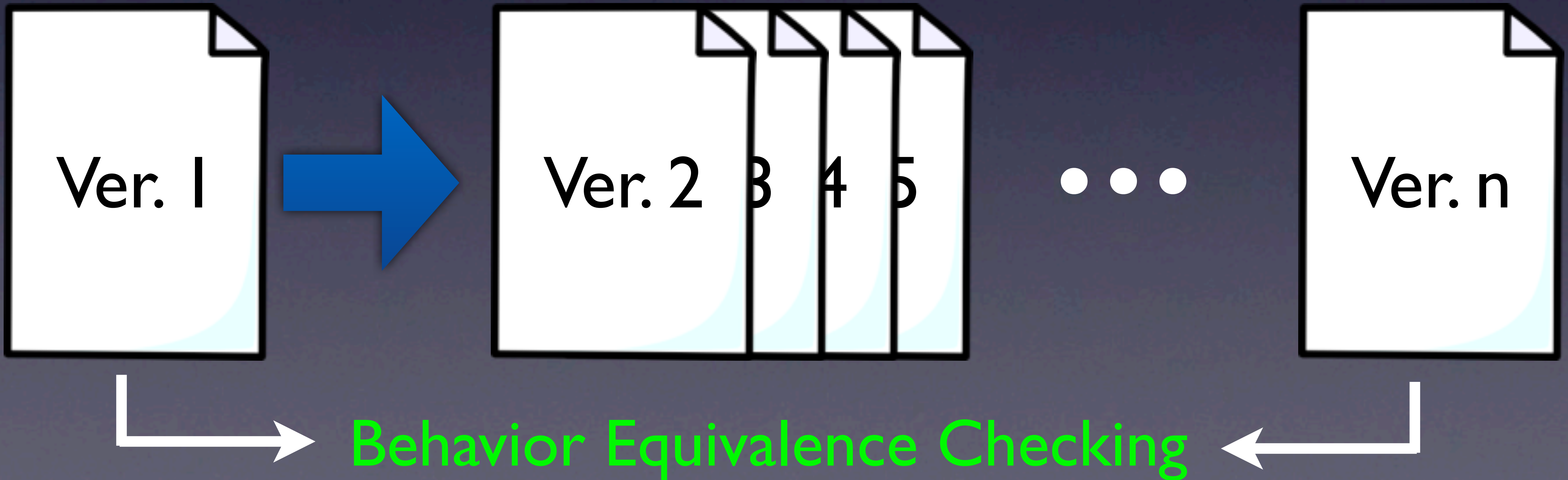
- Development of software for nuclear power plant reactor protection system (2000 ~ 2007)
- To be deployed in Shin-Uljin NPP in Korea

Goals

- Use proven-effective formal techniques when feasible to demonstrate high reliability
- Domain experts should NOT be left alone in the dark

KNICS and VIS

- In implementation, code optimization is unavoidable
  - Demonstration of behavioral equivalence is critical
  - Testing is not sufficient
  - ➡ VIS formally checks behavior equivalence
  - Counterexample is provided when targets are inequivalent



KNICS and VIS

- KNICS uses FBD as implementation language
- VIS can accept Verilog code as an input
- FBD and Verilog have similar semantics

➡ Synthesis of Verilog from FBD is straightforward*

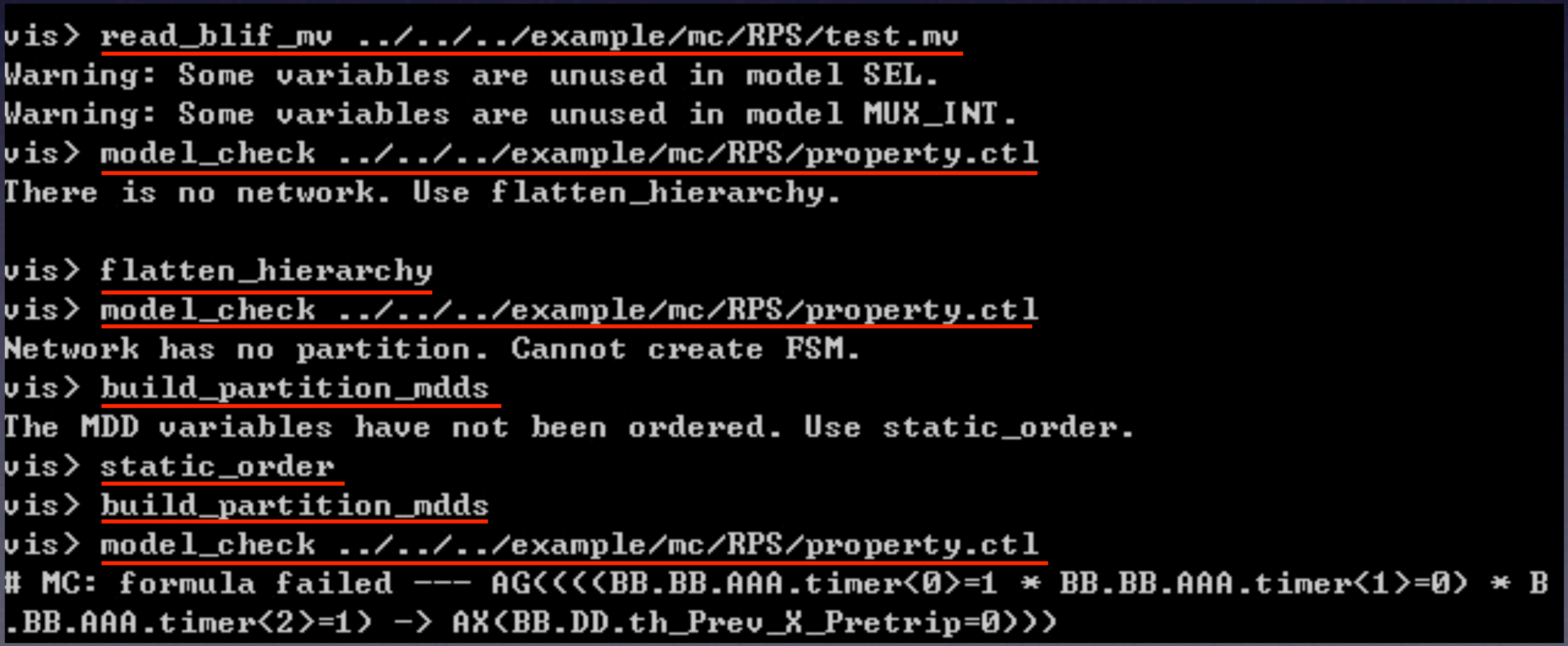
➡ VIS can perform CTL & LTL model checking

✓ I can say that VIS is worth for "★★★★☆"

*J.B. Yoo, S.D. Cha, and E.K. Jee, "*A Verification Framework for FBD based Software in Nuclear Power Plants*," APSEC 2008

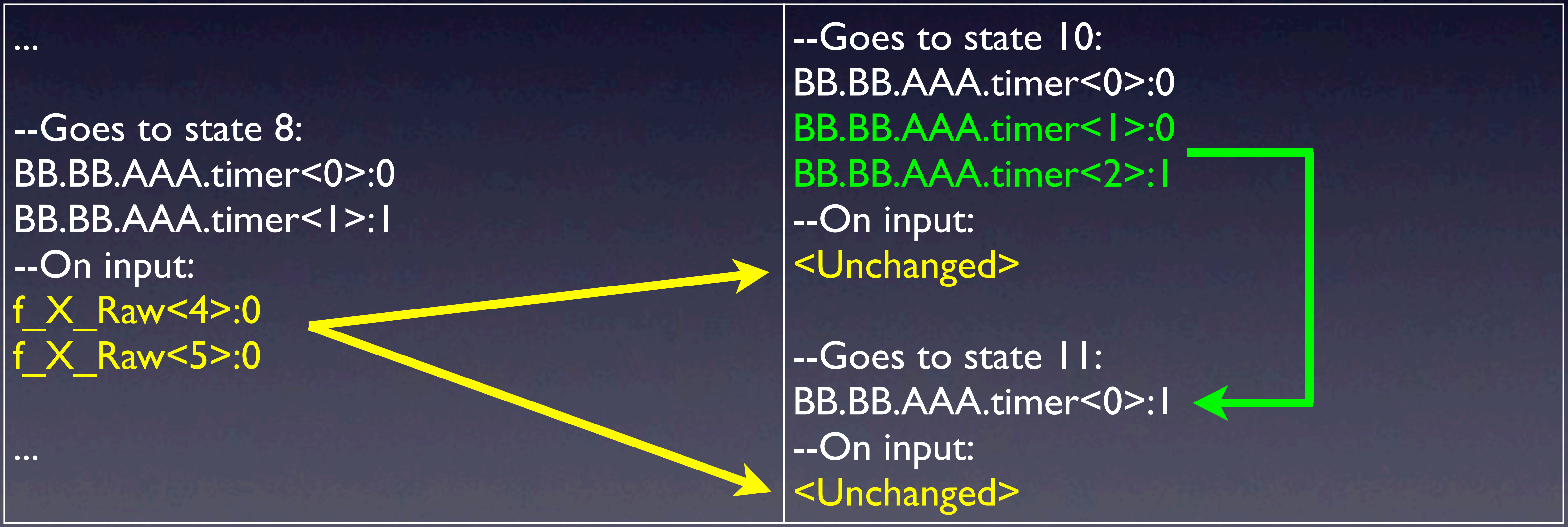
Practical Limitations on VIS

- Unfamiliar user interface
- Overly detailed verification process



```
vis> read_blif_mv ../../../example/mc/RPS/test.mv
Warning: Some variables are unused in model SEL.
Warning: Some variables are unused in model MUX_INT.
vis> model_check ../../../example/mc/RPS/property.ctl
There is no network. Use flatten_hierarchy.

vis> flatten_hierarchy
vis> model_check ../../../example/mc/RPS/property.ctl
Network has no partition. Cannot create FSM.
vis> build_partition_mdds
The MDD variables have not been ordered. Use static_order.
vis> static_order
vis> build_partition_mdds
vis> model_check ../../../example/mc/RPS/property.ctl
# MC: formula failed --- AG((((BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0) * B
.BB.AAA.timer<2>=1) -> AX(BB.DD.th_Prev_X_Pretrip=0)))
```

- Information partialities in counterexamples decrease readability
  - Unchanged input & state values are not shown
  - Output values are not shown in equivalence checking



```
...

--Goes to state 8:
BB.BB.AAA.timer<0>:0
BB.BB.AAA.timer<1>:1
--On input:
f_X_Raw<4>:0
f_X_Raw<5>:0

...
```

```
--Goes to state 10:
BB.BB.AAA.timer<0>:0
BB.BB.AAA.timer<1>:0
BB.BB.AAA.timer<2>:1
--On input:
<Unchanged>

--Goes to state 11:
BB.BB.AAA.timer<0>:1
--On input:
<Unchanged>
```

- Textual display is not adequate for counterexample representation

```
vis release 2.0
vis> read_blif_mv a.mv
vis> flatten_hierarchy
vis> seq_verify b.mv
--State 0:
state$NTK2:S1
state:S0
th_Prev_X_Pretrip$NTK2:1
th_Prev_X_Pretrip:1
timer$NTK2:T0
timer:T0

--Goes to state 1:
state:S1
timer$NTK2:T1
timer:T1
--On input:
f_X<0>:0
f_X<1>:1
f_X<2>:1
f_X<3>:1
f_X<4>:1
f_X<5>:0
f_X<6>:1
--Goes to state 2:
timer$NTK2:T2
timer:T2
--On input:
<Unchanged>

--Goes to state 3:
timer$NTK2:T3
timer:T3
--On input:
<Unchanged>

--Goes to state 4:
timer$NTK2:T4
timer:T4
--On input:
<Unchanged>

--Goes to state 5:
timer$NTK2:T5
timer:T5
--On input:
<Unchanged>
--Goes to state 6:
state$NTK2:S0
state:S2
th_Prev_X_Pretrip$NTK2:0
th_Prev_X_Pretrip:0
--On input:
<Unchanged>

--Goes to state 7:
timer$NTK2:T0
timer:T0
--On input:
f_X<3>:0
f_X<6>:0

Networks are NOT sequentially
equivalent.
```

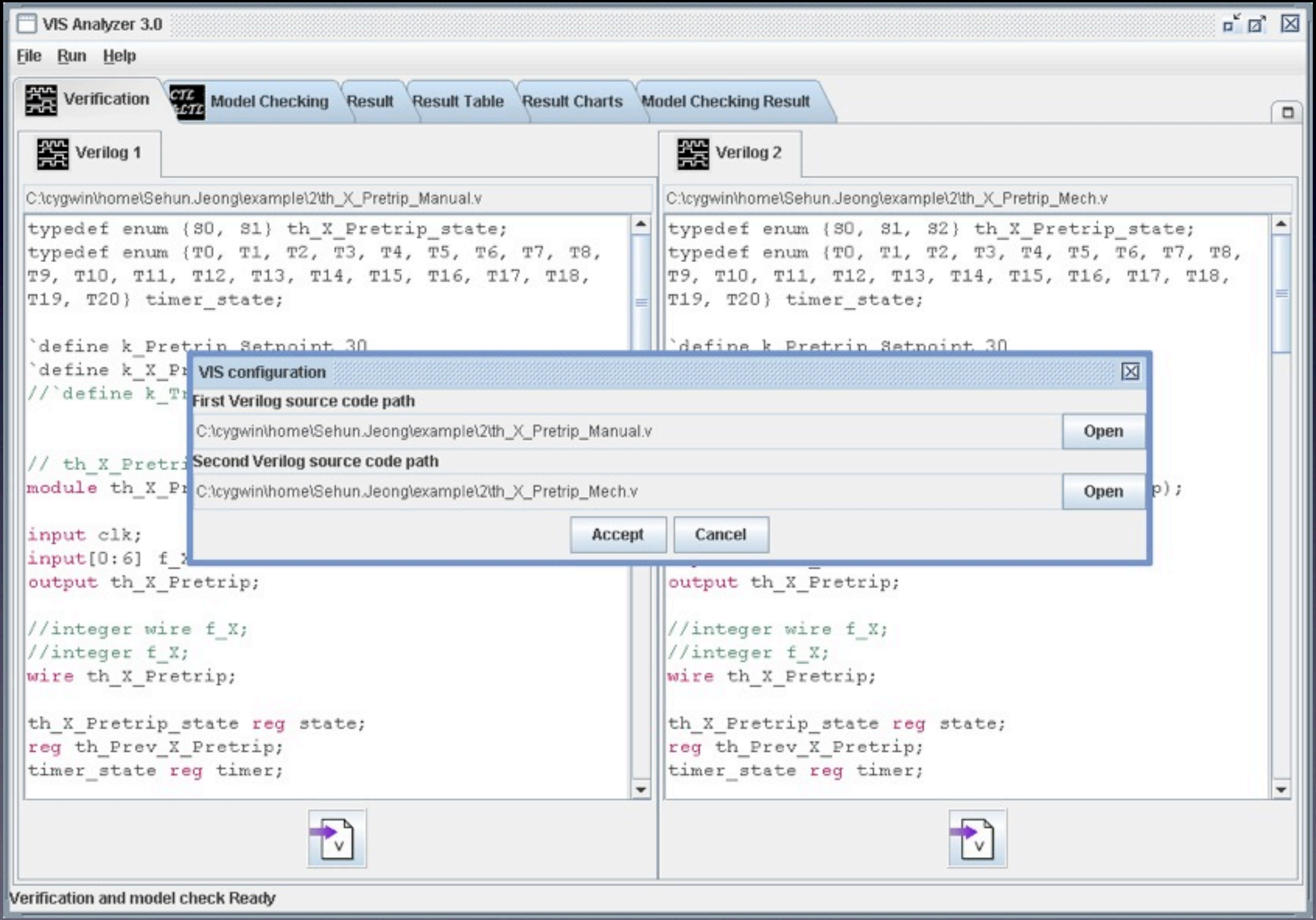
Outline

- Motivation
- Our Solution
- Comparisons
- Conclusions & Future Work

VIS Analyzer 3.0

- Graphic user interface
- Automation of VIS verification features
  - Equivalence checking
  - Model checking
  - Simulation
- Verification Results without partialities
- Visualization of the results

Side-by-side code comparing with syntax highlighting



C:\cygwin\home\Sehun.Jeong\example\2\th_X_Pretrip_Manual.v

```
typedef enum {S0, S1} th_X_Pretrip_state;
typedef enum {T0, T1, T2, T3, T4, T5, T6, T7, T8,
T9, T10, T11, T12, T13, T14, T15, T16, T17, T18,
T19, T20} timer_state;

`define k_Pretrip_Setpoint 30
`define k_X_Pr
//`define k_Tr


// th_X_Pretri
module th_X_Pr

input clk;
input[0:6] f_X
output th_X_Pretrip;

//integer wire f_X;
//integer f_X;
wire th_X_Pretrip;

th_X_Pretrip_state reg state;
reg th_Prev_X_Pretrip;
timer_state reg timer;
```

C:\cygwin\home\Sehun.Jeong\example\2\th_X_Pretrip_Mech.v

```
typedef enum {S0, S1, S2} th_X_Pretrip_state;
typedef enum {T0, T1, T2, T3, T4, T5, T6, T7, T8,
T9, T10, T11, T12, T13, T14, T15, T16, T17, T18,
T19, T20} timer_state;

`define k_Pretrip_Setpoint 30

p);

output th_X_Pretrip;

//integer wire f_X;
//integer f_X;
wire th_X_Pretrip;

th_X_Pretrip_state reg state;
reg th_Prev_X_Pretrip;
timer_state reg timer;
```

VIS configuration

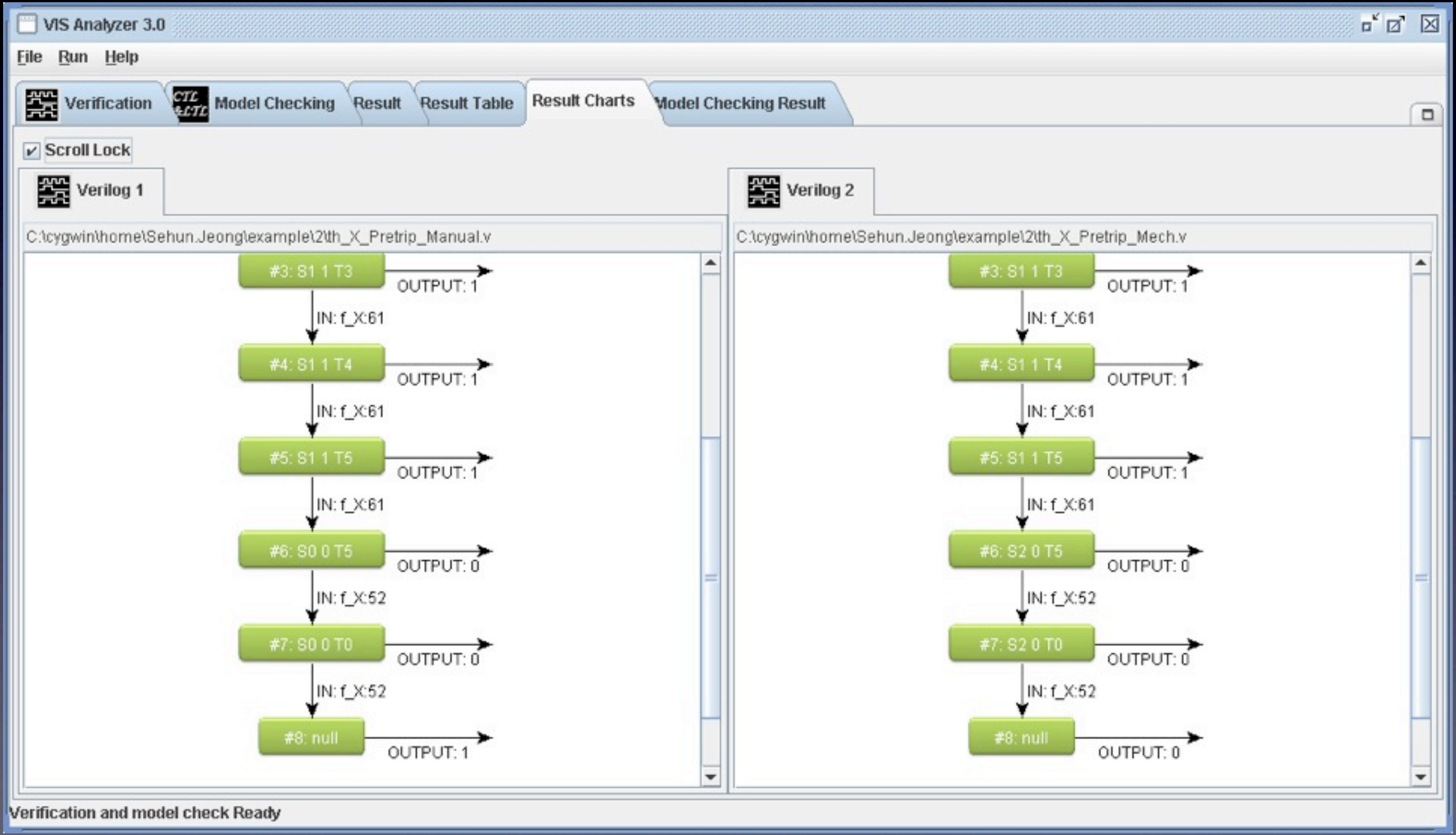
First Verilog source code path: C:\cygwin\home\Sehun.Jeong\example\2\th_X_Pretrip_Manual.v [Open]

Second Verilog source code path: C:\cygwin\home\Sehun.Jeong\example\2\th_X_Pretrip_Mech.v [Open]

[Accept] [Cancel]

Verification and model check Ready

Intuitive counterexample visualization



C:\cygwin\home\Sehun.Jeong\example\2\th_X_Pretrip_Manual.v

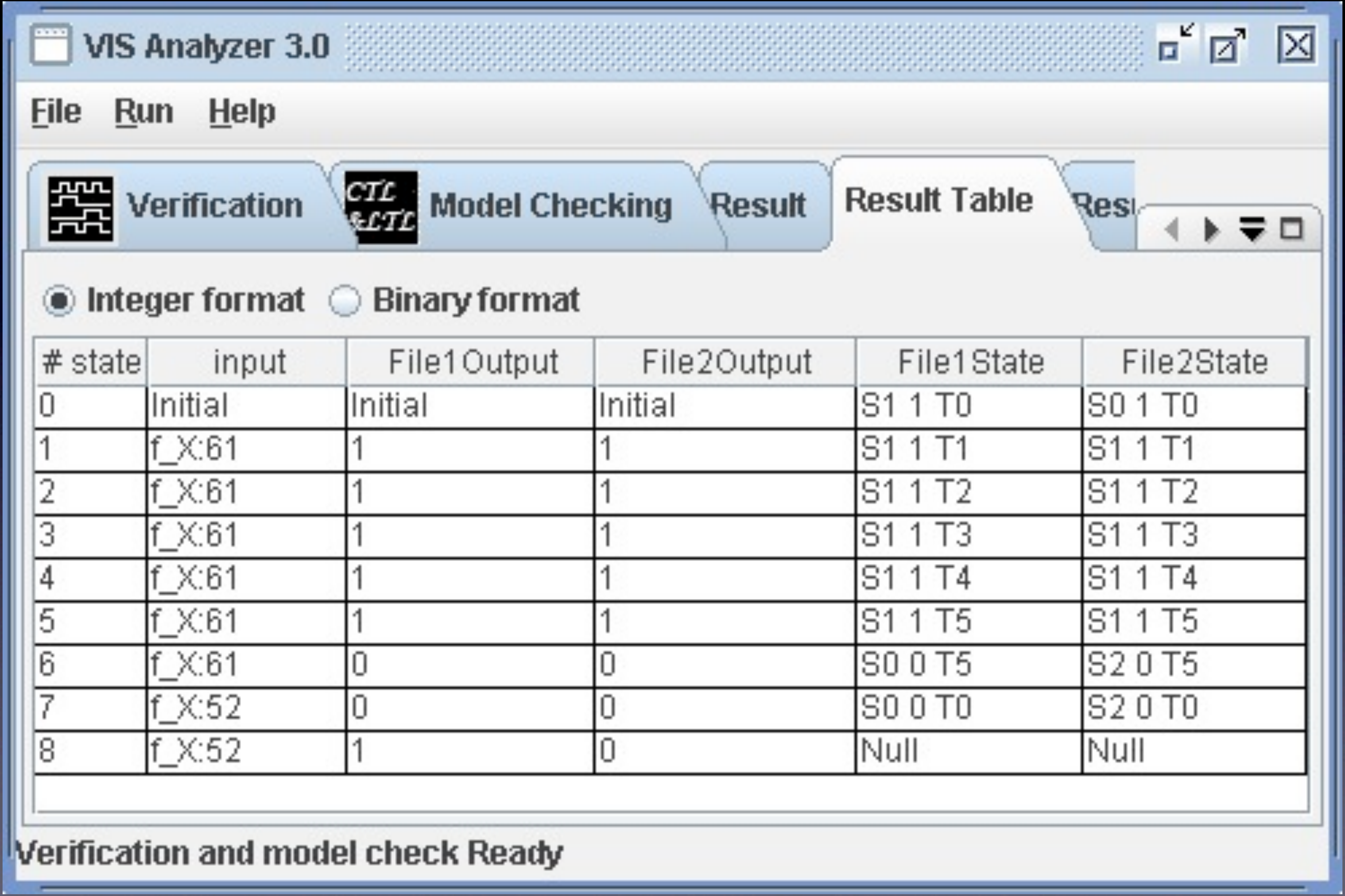
#3: S1 1 T3 — OUTPUT: 1
IN: f_X:61
#4: S1 1 T4 — OUTPUT: 1
IN: f_X:61
#5: S1 1 T5 — OUTPUT: 1
IN: f_X:61
#6: S0 0 T5 — OUTPUT: 0
IN: f_X:52
#7: S0 0 T0 — OUTPUT: 0
IN: f_X:52
#8: null — OUTPUT: 1

C:\cygwin\home\Sehun.Jeong\example\2\th_X_Pretrip_Mech.v

#3: S1 1 T3 — OUTPUT: 1
IN: f_X:61
#4: S1 1 T4 — OUTPUT: 1
IN: f_X:61
#5: S1 1 T5 — OUTPUT: 1
IN: f_X:61
#6: S2 0 T5 — OUTPUT: 0
IN: f_X:52
#7: S2 0 T0 — OUTPUT: 0
IN: f_X:52
#8: null — OUTPUT: 0

Verification and model check Ready

Flexible display of verification result

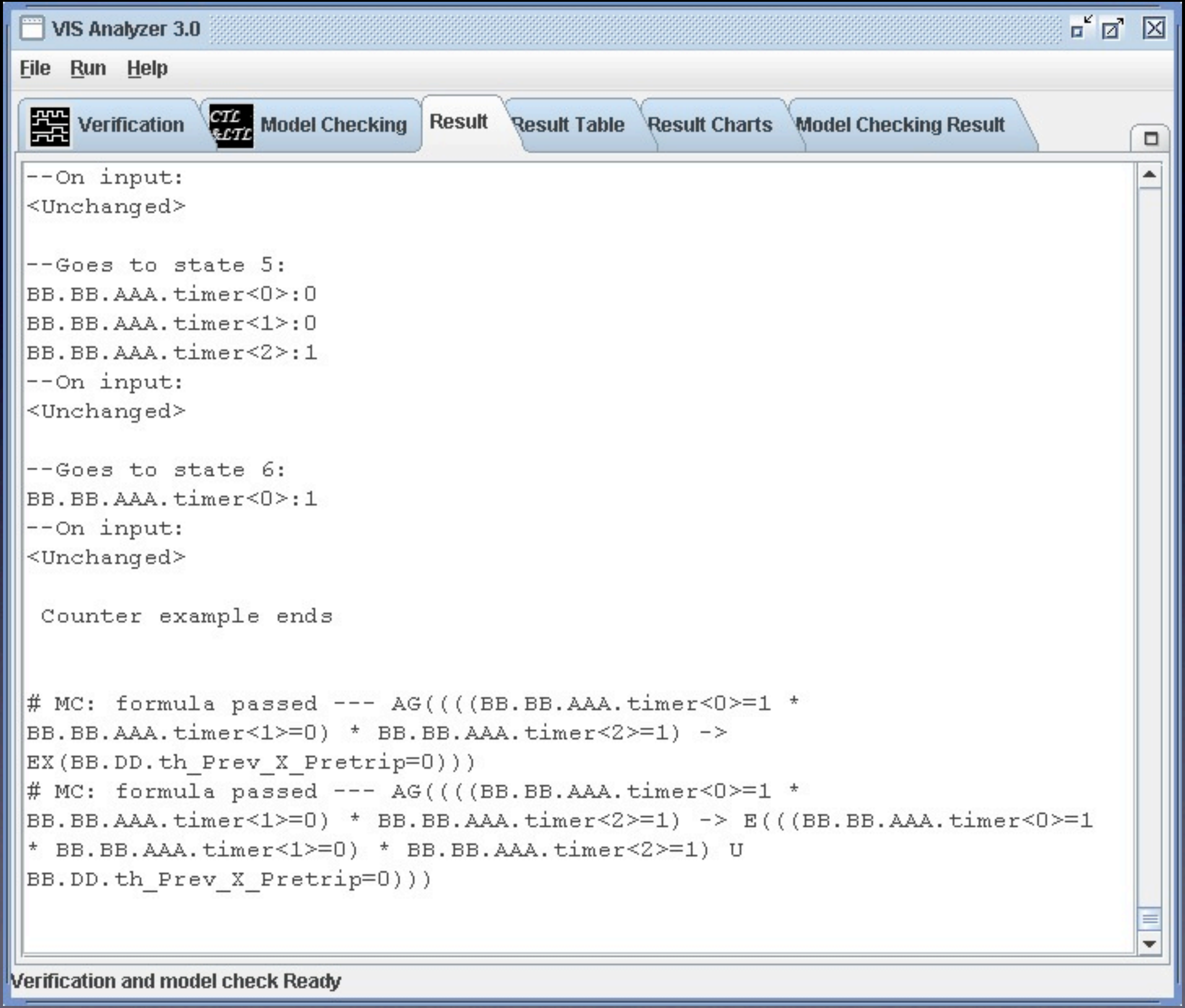


[● Integer format] ○ Binary format

# state	input	File1Output	File2Output	File1State	File2State
0	Initial	Initial	Initial	S1 1 T0	S0 1 T0
1	f_X:61	1	1	S1 1 T1	S1 1 T1
2	f_X:61	1	1	S1 1 T2	S1 1 T2
3	f_X:61	1	1	S1 1 T3	S1 1 T3
4	f_X:61	1	1	S1 1 T4	S1 1 T4
5	f_X:61	1	1	S1 1 T5	S1 1 T5
6	f_X:61	0	0	S0 0 T5	S2 0 T5
7	f_X:52	0	0	S0 0 T0	S2 0 T0
8	f_X:52	1	0	Null	Null

Verification and model check Ready

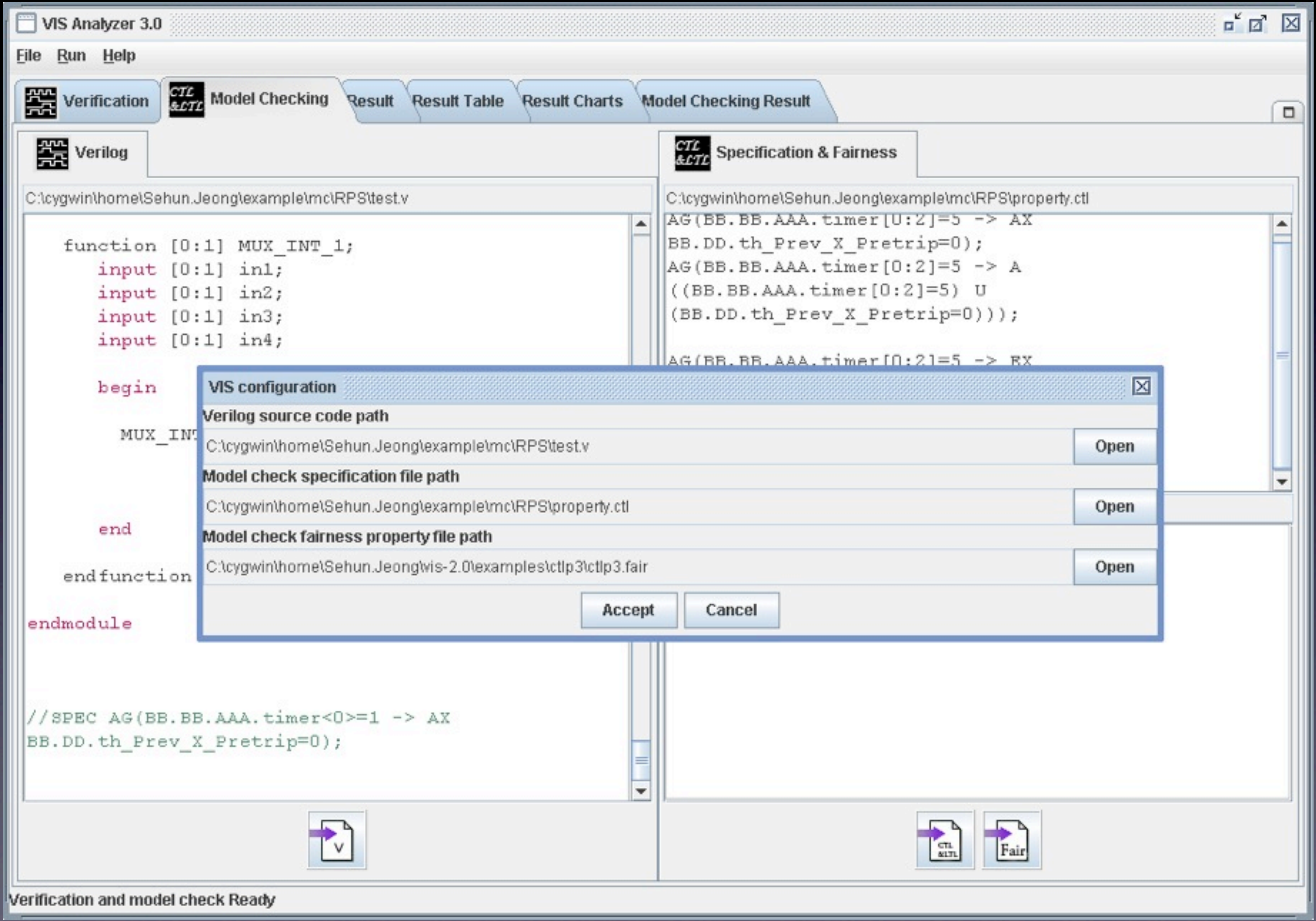
Raw data for reference



```
--On input:
<Unchanged>

--Goes to state 5:
BB.BB.AAA.timer<0>:0
BB.BB.AAA.timer<1>:0
BB.BB.AAA.timer<2>:1
--On input:
<Unchanged>

--Goes to state 6:
BB.BB.AAA.timer<0>:1
--On input:
<Unchanged>

 Counter example ends


# MC: formula passed --- AG((((BB.BB.AAA.timer<0>=1 *
BB.BB.AAA.timer<1>=0) * BB.BB.AAA.timer<2>=1) ->
EX(BB.DD.th_Prev_X_Pretrip=0)))
# MC: formula passed --- AG((((BB.BB.AAA.timer<0>=1 *
BB.BB.AAA.timer<1>=0) * BB.BB.AAA.timer<2>=1) -> E(((BB.BB.AAA.timer<0>=1
* BB.BB.AAA.timer<1>=0) * BB.BB.AAA.timer<2>=1) U
BB.DD.th_Prev_X_Pretrip=0)))
```

Verification and model check Ready

3-in-1 Model checking window



C:\cygwin\home\Sehun.Jeong\example\mc\RPS\test.v

```
   function [0:1] MUX_INT_1;
      input [0:1] in1;
      input [0:1] in2;
      input [0:1] in3;
      input [0:1] in4;

      begin

        MUX_INT

      end

   endfunction

endmodule



//SPEC AG(BB.BB.AAA.timer<0>=1 -> AX
BB.DD.th_Prev_X_Pretrip=0);
```

C:\cygwin\home\Sehun.Jeong\example\mc\RPS\property.ctl

```
AG(BB.BB.AAA.timer[0:2]=5 -> AX
BB.DD.th_Prev_X_Pretrip=0);
AG(BB.BB.AAA.timer[0:2]=5 -> A
((BB.BB.AAA.timer[0:2]=5) U
(BB.DD.th_Prev_X_Pretrip=0)));

AG(BB.BB.AAA.timer[0:2]=5 -> EX
```

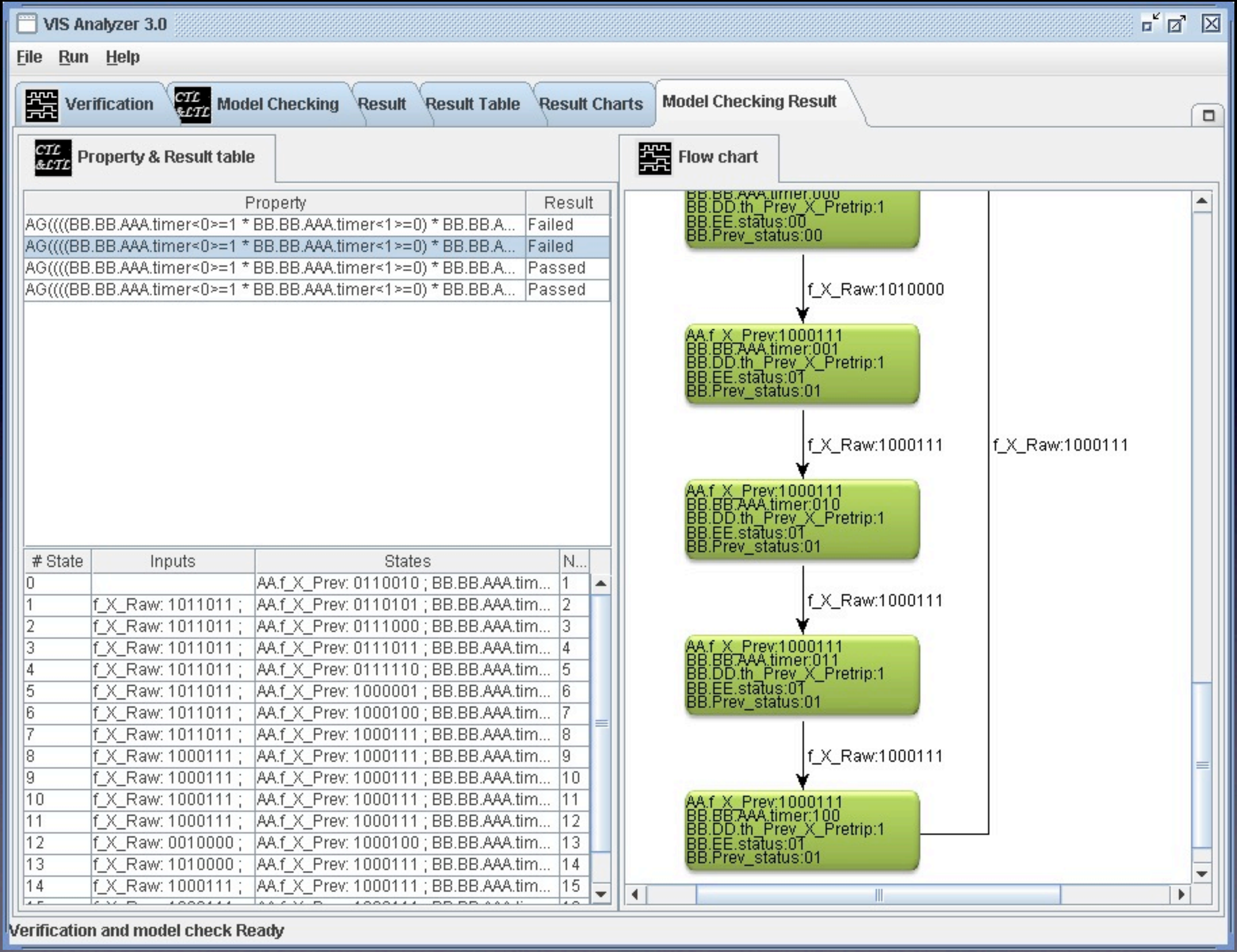
VIS configuration

Verilog source code path
C:\cygwin\home\Sehun.Jeong\example\mc\RPS\test.v

Model check specification file path
C:\cygwin\home\Sehun.Jeong\example\mc\RPS\property.ctl

Model check fairness property file path
C:\cygwin\home\Sehun.Jeong\vis-2.0\examples\ctlp3\ctlp3.fair

Verification and model check Ready



VIS Analyzer 3.0

File Run Help

Verification | Model Checking | Result | Result Table | Result Charts | Model Checking Result

Property & Result table

Property	Result
AG((((BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0) * BB.BB.A...	Failed
AG((((BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0) * BB.BB.A...	Failed
AG((((BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0) * BB.BB.A...	Passed
AG((((BB.BB.AAA.timer<0>=1 * BB.BB.AAA.timer<1>=0) * BB.BB.A...	Passed

# State	Inputs	States	N...
0		AA.f_X_Prev: 0110010 ; BB.BB.AAA.tim...	1
1	f_X_Raw: 1011011 ;	AA.f_X_Prev: 0110101 ; BB.BB.AAA.tim...	2
2	f_X_Raw: 1011011 ;	AA.f_X_Prev: 0111000 ; BB.BB.AAA.tim...	3
3	f_X_Raw: 1011011 ;	AA.f_X_Prev: 0111011 ; BB.BB.AAA.tim...	4
4	f_X_Raw: 1011011 ;	AA.f_X_Prev: 0111110 ; BB.BB.AAA.tim...	5
5	f_X_Raw: 1011011 ;	AA.f_X_Prev: 1000001 ; BB.BB.AAA.tim...	6
6	f_X_Raw: 1011011 ;	AA.f_X_Prev: 1000100 ; BB.BB.AAA.tim...	7
7	f_X_Raw: 1011011 ;	AA.f_X_Prev: 1000111 ; BB.BB.AAA.tim...	8
8	f_X_Raw: 1000111 ;	AA.f_X_Prev: 1000111 ; BB.BB.AAA.tim...	9
9	f_X_Raw: 1000111 ;	AA.f_X_Prev: 1000111 ; BB.BB.AAA.tim...	10
10	f_X_Raw: 1000111 ;	AA.f_X_Prev: 1000111 ; BB.BB.AAA.tim...	11
11	f_X_Raw: 1000111 ;	AA.f_X_Prev: 1000111 ; BB.BB.AAA.tim...	12
12	f_X_Raw: 0010000 ;	AA.f_X_Prev: 1000100 ; BB.BB.AAA.tim...	13
13	f_X_Raw: 1010000 ;	AA.f_X_Prev: 1000111 ; BB.BB.AAA.tim...	14
14	f_X_Raw: 1000111 ;	AA.f_X_Prev: 1000111 ; BB.BB.AAA.tim...	15

Flow chart

```
BB.BB.AAA.timer:000
BB.DD.th_Prev_X_Pretrip:1
BB.EE.status:00
BB.Prev_status:00
        |  f_X_Raw:1010000
        v
AA.f_X_Prev:1000111
BB.BB.AAA.timer:001
BB.DD.th_Prev_X_Pretrip:1
BB.EE.status:01
BB.Prev_status:01
        |  f_X_Raw:1000111                f_X_Raw:1000111
        v
AA.f_X_Prev:1000111
BB.BB.AAA.timer:010
BB.DD.th_Prev_X_Pretrip:1
BB.EE.status:01
BB.Prev_status:01
        |  f_X_Raw:1000111
        v
AA.f_X_Prev:1000111
BB.BB.AAA.timer:011
BB.DD.th_Prev_X_Pretrip:1
BB.EE.status:01
BB.Prev_status:01
        |  f_X_Raw:1000111
        v
AA.f_X_Prev:1000111
BB.BB.AAA.timer:100
BB.DD.th_Prev_X_Pretrip:1
BB.EE.status:01
BB.Prev_status:01
```

Verification and model check Ready

Outline

- Motivation
- Our Solution
- Comparisons
- Conclusions & Future Work

Comparison 1

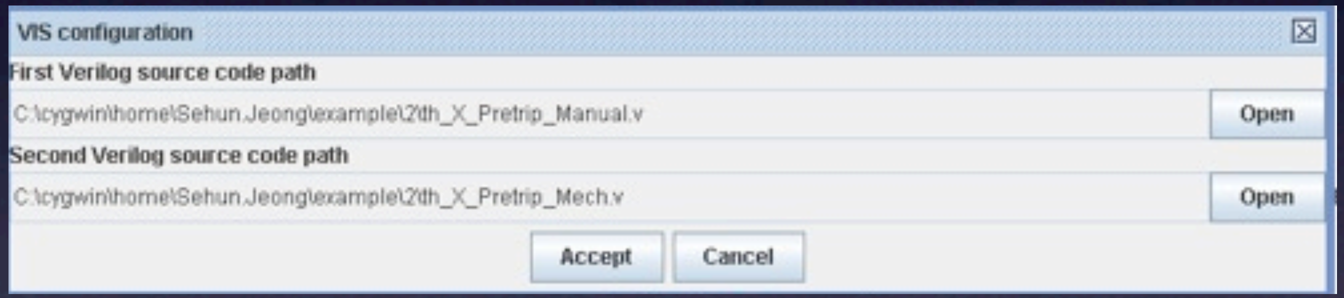
- Equivalence checking process

```
c:\>vl2mv th_X_Pretrip_Manual.v
c:\>vl2mv th_X_Pretrip_Mach.v

vis release 2.0 (compiled Sat Jun 14 12: ... )

vis> read_blif_mv th_X_Pretrip_Manual.mv
vis> flatten_hierarchy
vis> seq_verify th_X_Pretrip_Mech.mv
--State 0:
state$NTK2:S1


...
```

Comparison 1

- Equivalence checking process

```
c:\>vl2mv th_X_Pretrip_Manual.v
c:\>vl2mv th_X_Pretrip_Mach.v

vis release 2.0 (compiled Sat Jun 14 12: ... )

vis> read_blif_mv th_X_Pretrip_Manual.mv
vis> flatten_hierarchy
vis> seq_verify th_X_Pretrip_Mech.mv
--State 0:
state$NTK2:S1

...
```



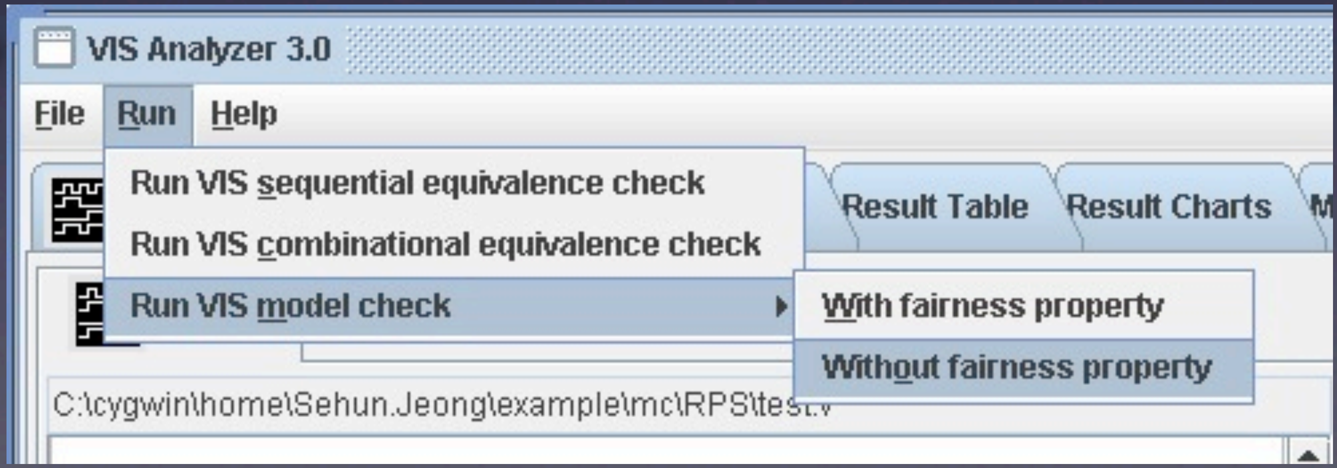
VIS configuration

First Verilog source code path

C:\cygwin\home\Sehun.Jeong\example\2\th_X_Pretrip_Manual.v

Second Verilog source code path

C:\cygwin\home\Sehun.Jeong\example\2\th_X_Pretrip_Mech.v



VIS Analyzer 3.0
File
Run
Help
Run VIS sequential equivalence check
Run VIS combinational equivalence check
Run VIS model check
With fairness property
Without fairness property
Result Table
Result Charts

C:\cygwin\home\Sehun.Jeong\example\mc\RPS\test.v

- Equivalence checking result

```
--Goes to state 6:
state$NTK2:S0
state:S2
th_Prev_X_Pretrip$NTK2:0
th_Prev_X_Pretrip:0
--On input:
<Unchanged>

--Goes to state 7:
timer$NTK2:T0
timer:T0
--On input:
f_X<3>:0
f_X<6>:0
```

• Equivalence checking result

--Goes to state 6:
state$NTK2:S0
state:S2
th_Prev_X_Pretrip$NTK2:0
th_Prev_X_Pretrip:0
--On input:
<Unchanged>

--Goes to state 7:
timer$NTK2:T0
timer:T0
--On input:
f_X<3>:0
f_X<6>:0

f_X<0>:0	**<State 6>**
f_X<1>:1	state$NTK2:S0
f_X<2>:1	state:S2
f_X<3>:1	th_Prev_X_Pretrip$NTK2:0
f_X<4>:1	th_Prev_X_Pretrip:0
f_X<5>:0	timer$NTK2:T5
f_X<6>:1	timer:T5
th_X_Pretrip: 0	

f_X<0>:0	**<State 7>**
f_X<1>:1	state$NTK2:S0
f_X<2>:1	state:S2
f_X<3>:0	th_Prev_X_Pretrip$NTK2:0
f_X<4>:1	th_Prev_X_Pretrip:0
f_X<5>:0	timer$NTK2:T0
f_X<6>:0	timer:T0
th_X_Pretrip: 0	

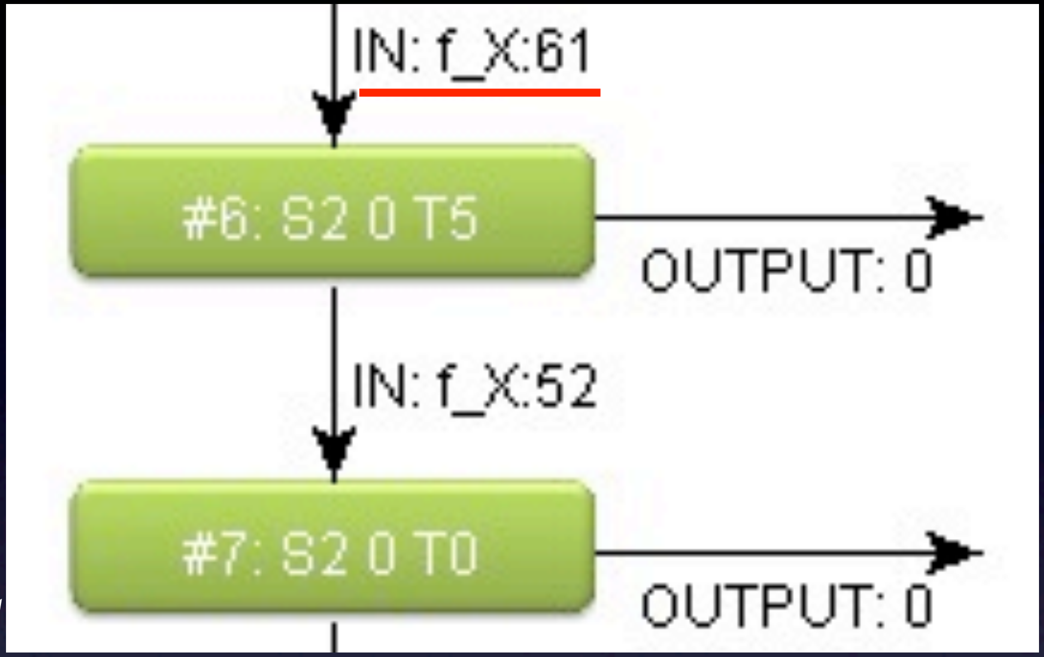
• Equivalence checking result representation

f_X<0>:0	**<State 6>**
f_X<1>:1	state$NTK2:S0
f_X<2>:1	state:S2
f_X<3>:1	th_Prev_X_Pretrip$NTK2:0
f_X<4>:1	th_Prev_X_Pretrip:0
f_X<5>:0	timer$NTK2:T5
f_X<6>:1	timer:T5
th_X_Pretrip: 0	

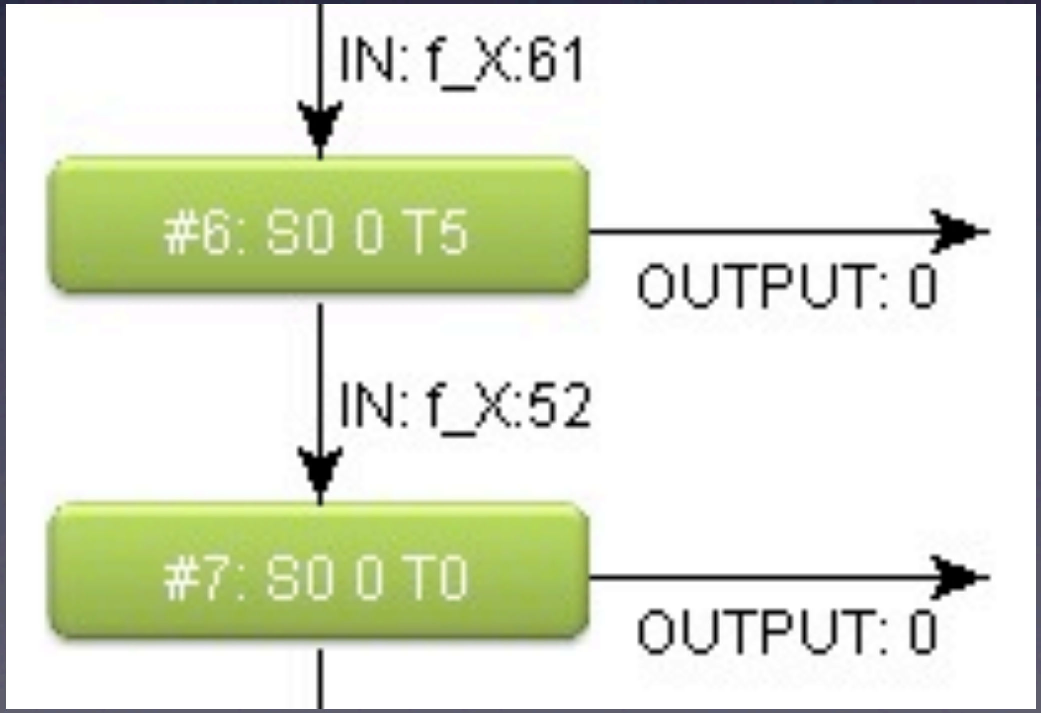
f_X<0>:0	**<State 7>**
f_X<1>:1	state$NTK2:S0
f_X<2>:1	state:S2
f_X<3>:0	th_Prev_X_Pretrip$NTK2:0
f_X<4>:1	th_Prev_X_Pretrip:0
f_X<5>:0	timer$NTK2:T0
f_X<6>:0	timer:T0
th_X_Pretrip: 0	

• Equivalence checking result representation

f_X<0>:0	**<State 6>**
f_X<1>:1	state$NTK2:S0
f_X<2>:1	state:S2
f_X<3>:1	th_Prev_X_Pretrip$NTK2:0
f_X<4>:1	th_Prev_X_Pretrip:0
f_X<5>:0	timer$NTK2:T5
f_X<6>:1	timer:T5
th_X_Pretrip: 0	



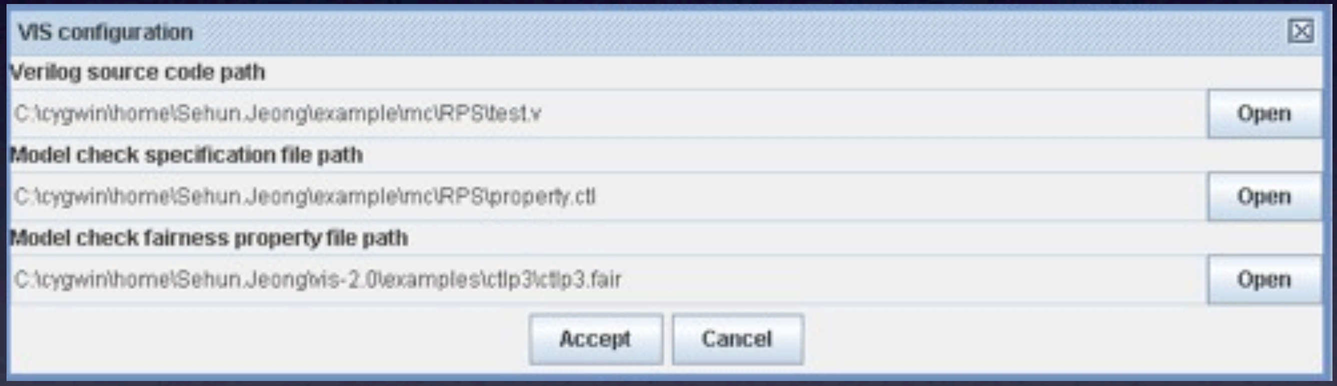
f_X<0>:0	**<State 7>**
f_X<1>:1	state$NTK2:S0
f_X<2>:1	state:S2
f_X<3>:0	th_Prev_X_Pretrip$NTK2:0
f_X<4>:1	th_Prev_X_Pretrip:0
f_X<5>:0	timer$NTK2:T0
f_X<6>:0	timer:T0
th_X_Pretrip: 0	



Comparison II

- Model checking result representation
  - Similar with equivalence checking

```
c:\>vl2mv test.v

vis release 2.0 (compiled Sat Jun 14 12: ... )

vis> read_blif_mv test.mv
vis> flatten_hierarchy
vis> static_order
vis> build_partition_mdds
vis> model_check -d 2 -i property.ctl
# MC: formula failed --- AG
((((BB.BB.AAA.timer<0>=1 *
...
```

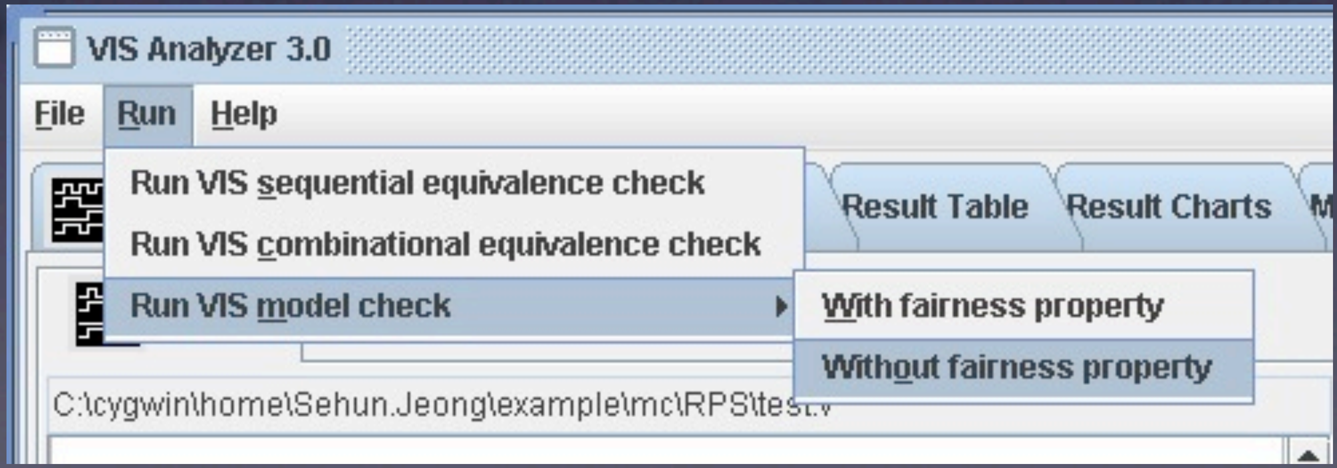


VIS configuration

Verilog source code path: C:\cygwin\home\Sehun.Jeong\example\mc\RPS\test.v

Model check specification file path: C:\cygwin\home\Sehun.Jeong\example\mc\RPS\property.ctl

Model check fairness property file path: C:\cygwin\home\Sehun.Jeong\vis-2.0\examples\ctlp3\ctlp3.fair

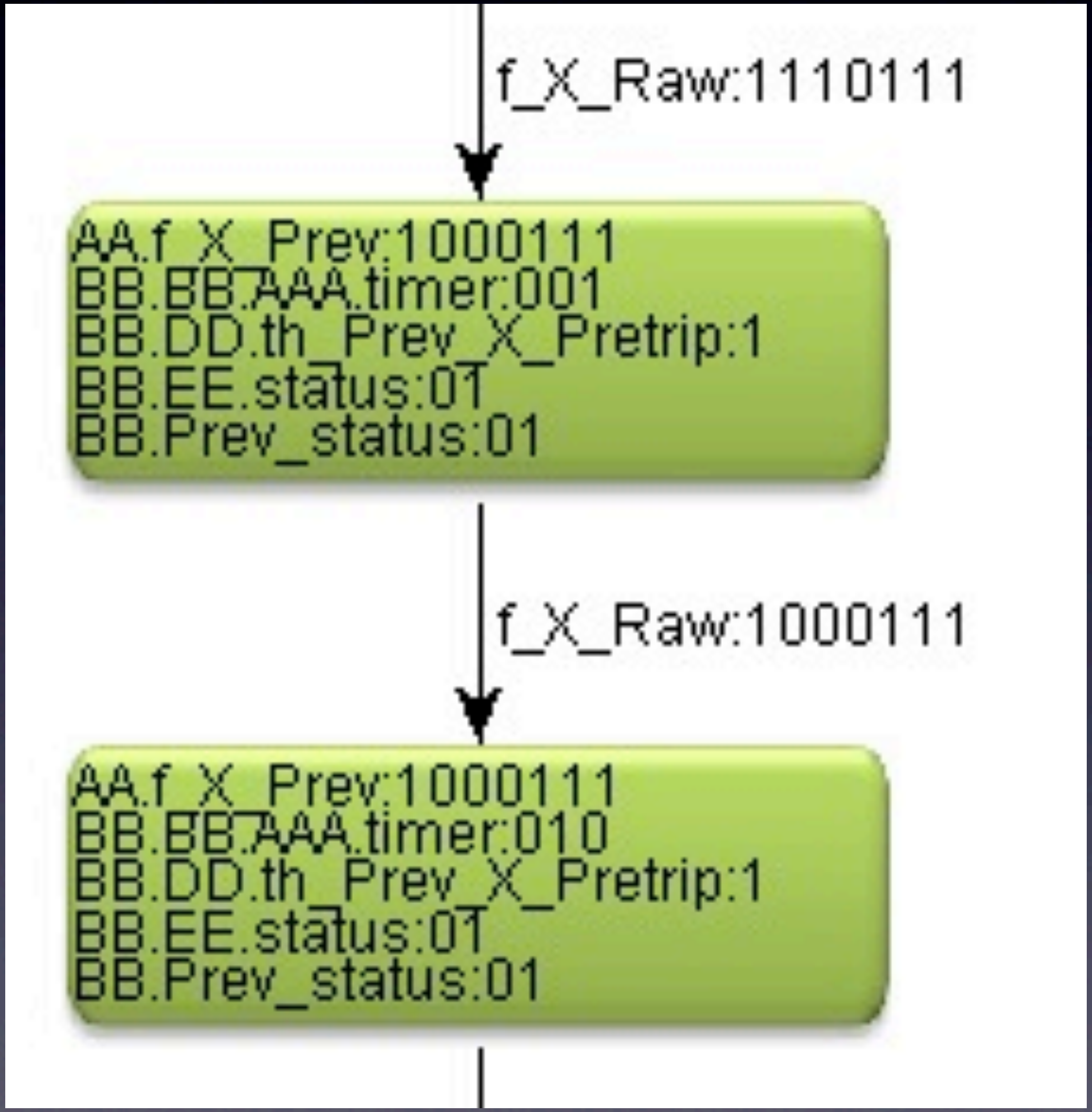


C:\cygwin\home\Sehun.Jeong\example\mc\RPS\test.v

VIS Analyzer 3.0
File
Run
Help
Run VIS sequential equivalence check
Run VIS combinational equivalence check
Run VIS model check
With fairness property
Without fairness property
Result Table
Result Charts

- Model checking result representation
  - Similar with equivalence checking

```
--Goes to state 7:
AA.f_X_Prev<0>:1
AA.f_X_Prev<1>:1
BB.BB.AAA.timer<0>:1
BB.EE.status<0>:1
BB.Prev_status<0>:1
--On input:
<Unchanged>

--Goes to state 8:
BB.BB.AAA.timer<0>:0
BB.BB.AAA.timer<1>:1
--On input:
f_X_Raw<4>:0
f_X_Raw<5>:0
```



VIS Analyzer really works well?

- Did not apply on a real project
  - The KNICS project is ended before the completion of the VIS Analyzer development
- But appears promising
- Any suggestions will be welcomed

Outline

- Motivation
- Our Solution
- Comparisons
- Conclusions & Future Work

Conclusions & Future Work

- Assistant tool for domain experts
- VIS Analyzer enhances usability of the VIS with;
  - GUI
  - Automation of verification process
  - Visualization of full verification result
- Currently focusing on more intuitive & informative display methods

Thank you