KNS 2013 Spring
광주 김대중컨벤션센터
2013.5.29 ~ 2013.5.31

Equivalence Checking between Pre-synthesis and Post-synthesis Programs by Using VIS

Jong-Hoon Lee
Dependable Software Laboratory
Konkuk University

KU KONKUK UNIVERSITY
DEPENDABLE SOFTWARE LABORATORY

Contents

1 Introduction

2 Verifications in FPGA

3 Equivalence Checking with VIS

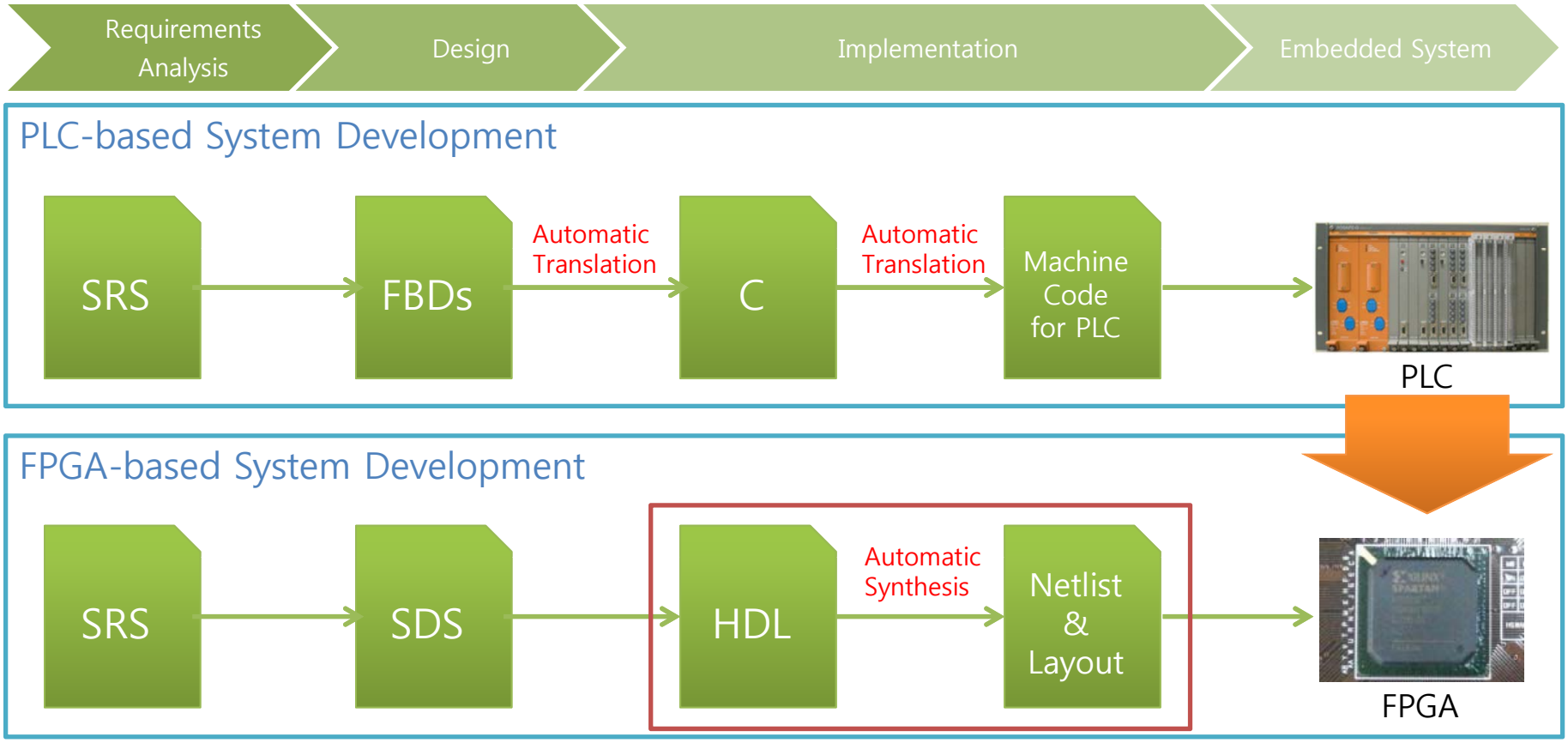
4 EDIFtoBLIF-MV

5 Case Study

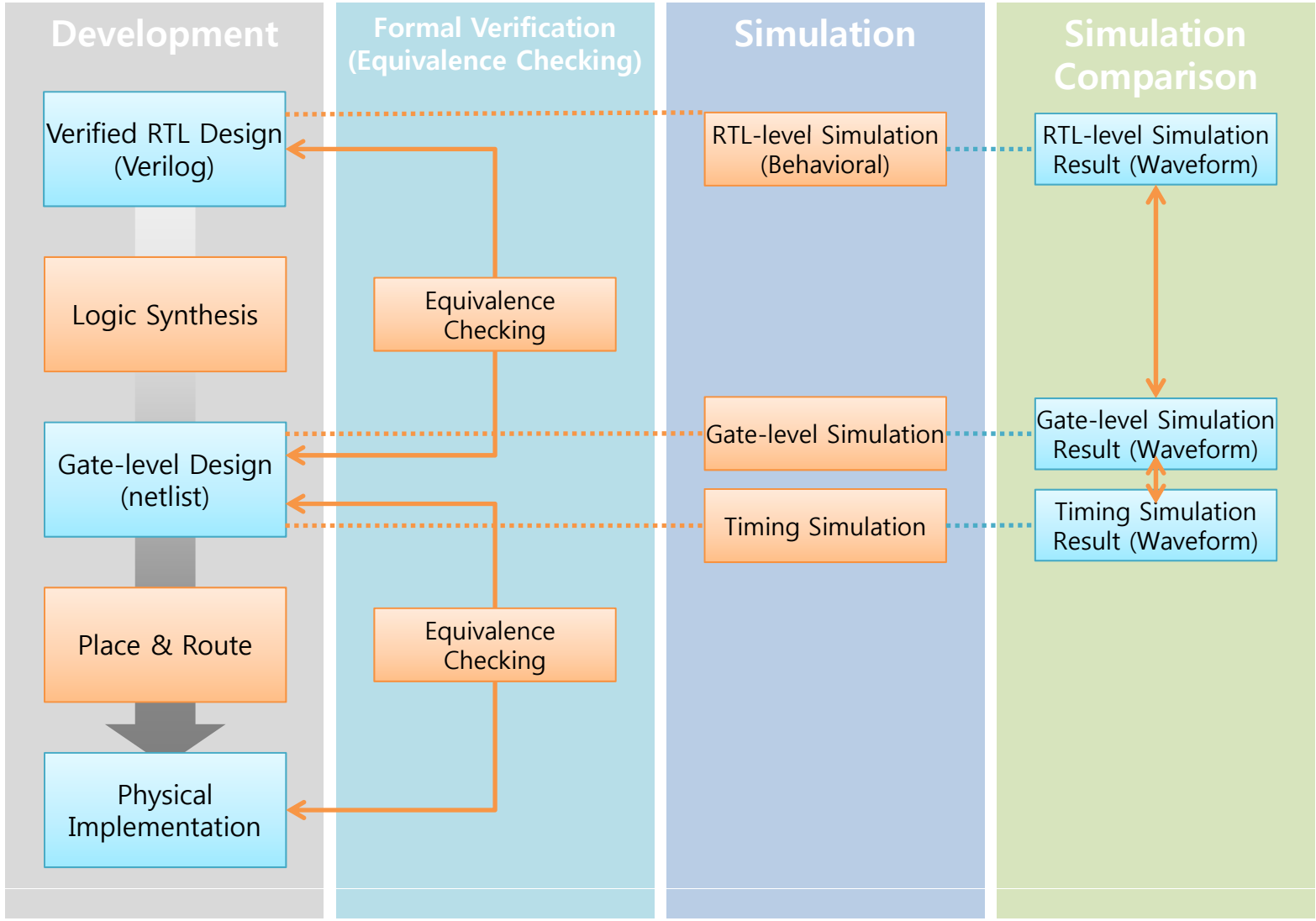
6 Conclusion and Future Work

Introduction

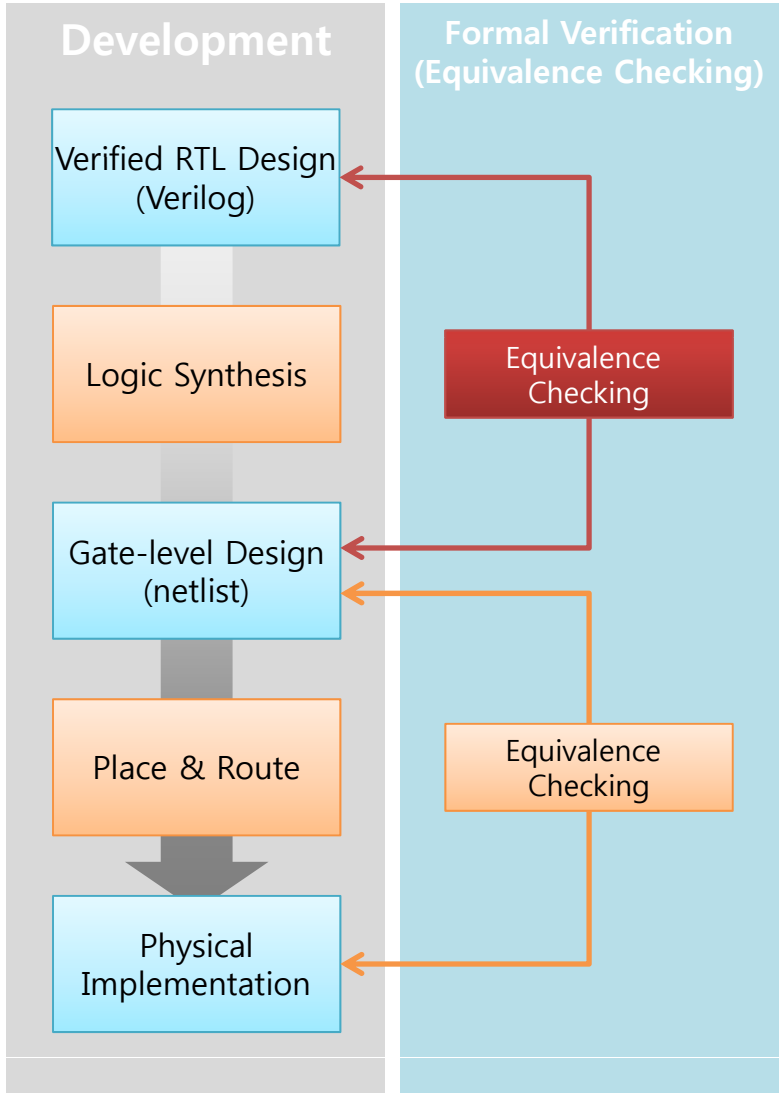
- Platform Change from PLC to FPGA in Nuclear Industry
  - Behaviorally Equivalence between Outputs is required
    - Formal Method based Techniques



Verifications in FPGA

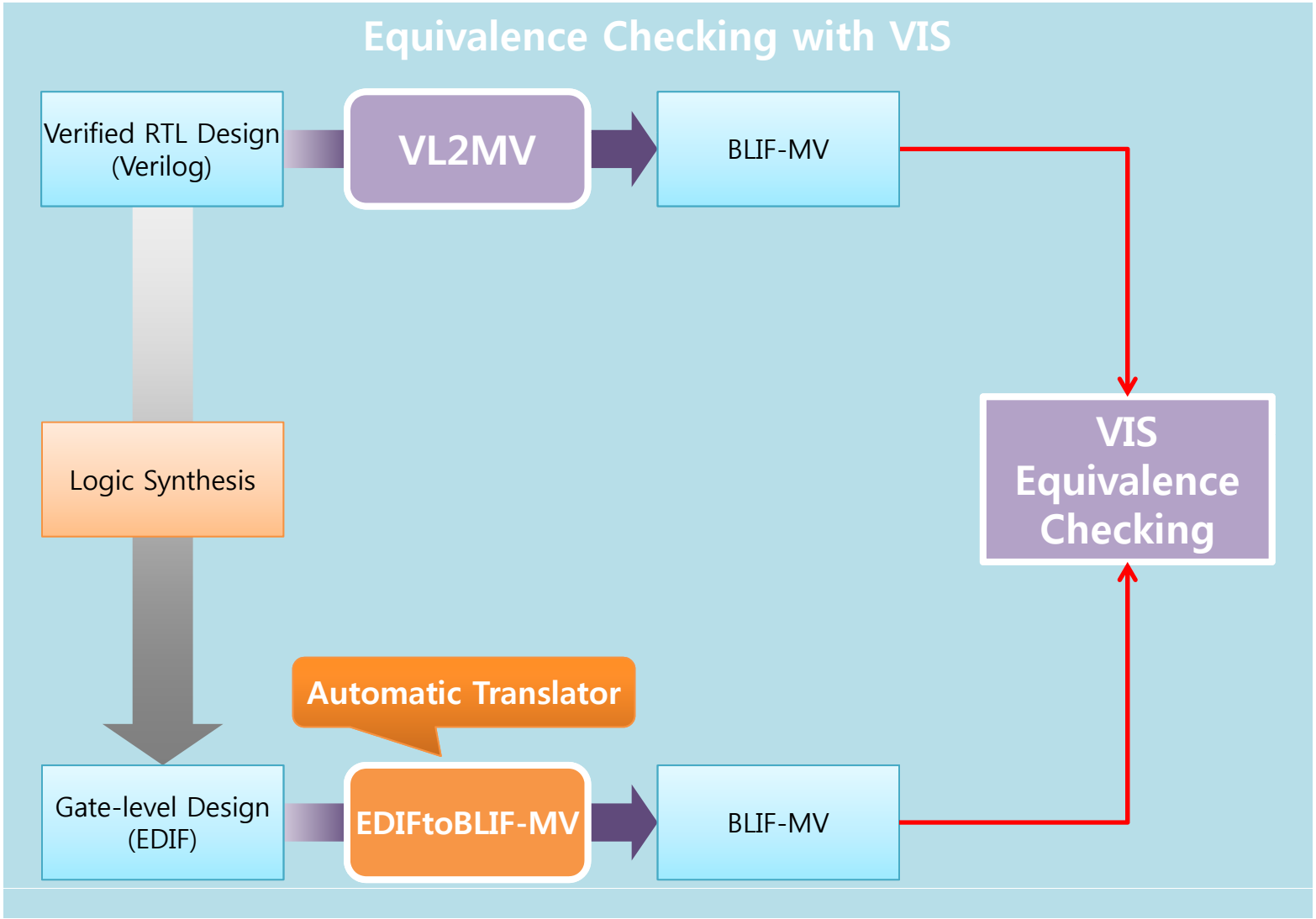


Equivalence Checking with VIS



- Equivalence Checking is required
  - Equivalence Checking between Designs
- VIS Verification System
  - Developed by Univ. of California at Berkeley, Univ. of Colorado at Boulder and Univ. of Texas
  - It has Equivalence Checking feature
  - It reads BLIF-MV modeling language
  - It only supports Verilog format

Equivalence Checking with VIS



EDIFtoBLIF-MV

- Translation Rule
  - We defined translation templates
  - Translate into BLIF-MV from EDIF using defined rules

	EDIF	BLIF-MV	Description
1-1	(cell <name_of_cell> …)	.model <name_of_cell> … .end	Cell in work library Translate to '.model'
1-2	(cell <name_of_cell> (view (interface (port <name_of_port> (direction <Input/Output>) (property function (<functionality>)) …	.names <Input> … <Output> .def 0 <Truth_table_of_functionality>	'combinational' cell in external library Translate to truth table
1-3	(cell <name_of_cell> (property is_sequential (1)) (view (interface (port <name_of_port> (direction <Input/Output>) (property function (<functionality>)) …	.r <Reset> .def 0 .latch <Input> <Output>	'sequential' cell in external library Translate to '.latch'
2	(cell <name_of_cell> (view (interface (port <name_of_port> (direction <Input/Output>)) …	.model <name_of_cell> .inputs <Input> <Input> … .outputs <Output> <Output> … … .end	Ports in Cell in work library Translate to '.inputs' and/or 'outputs'
3-1	(cell (view (contents (instance <name_of_insance> (viewRef <name_of_referred_view> (cellRef <name_of_referred_cell> …	.names <Input> … <Output> .def 0 <Truth_table_of_functionality_of_referred_cell>	Instance which refer to 'combinational' cell
3-2	(cell (view (contents (instance <name_of_instance> (viewRef <name_of_referred_view> (cellRef <name_of_referred_cell> …	.r <Reset> .def 0 .latch <Input> <Output>	Instance which refer to 'sequential' cell
3-3	(cell (view (contents (instance <name_of_instance> (viewRef <name_of_referred_view> (cellRef <name_of_referred_cell> …	.subckt <name_of_referred_cell> <name_of_instance> <Input> … <Output> … .end	Instance which refer to other cell in work library
3-4	(cell (view (contents (net <name_of_net> (joined (portRef <Input_Port> (instanceRef <name_of_Input_Instance>)) … (portRef <Output_Port> (instanceRef <name_of_Output_Instance>)) …	.names <name_of_Input_Instance> <name_of_Output_Instance> .def 0 1 1	Network connection between cells

EDIFtoBLIF-MV

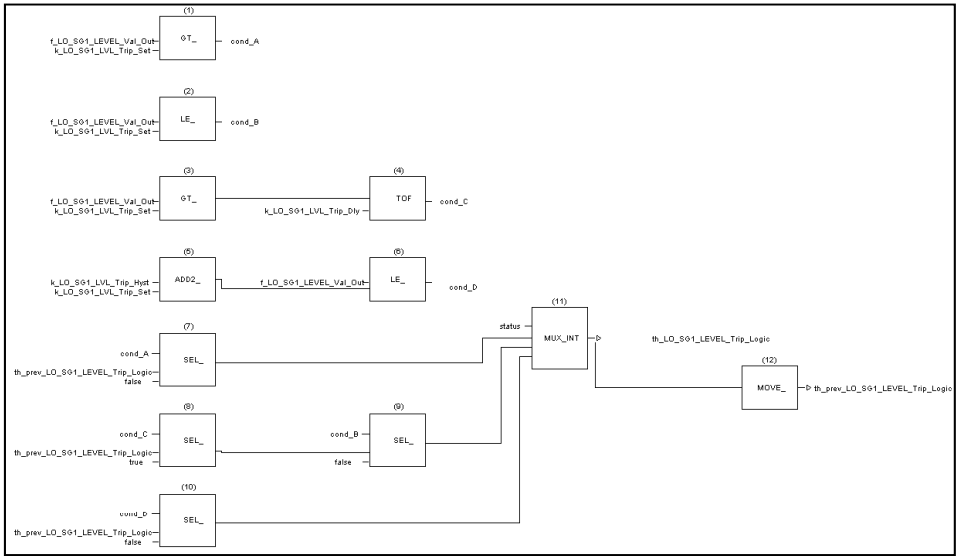
- A part of translation rule

Rule	EDIF	BLIF-MV	Description
1-1	(cell <name_of_cell> … …	.model <name_of_cell> … .end	Cell in work library Translate to '.model'
1-2	(cell <name_of_cell> (view (interface (port <name_of_port> (direction <Input/Output>) (property function (<functionality>)) …	.names <Input> … <Output> .def 0 <Truth_table_of_functionality>	'combinational' cell in external library Translate to truth table
1-3	(cell <name_of_cell> (property is_sequential (1)) (view (interface (port <name_of_port> (direction <Input/Output>) (property function (<functionality>)) …	.r <Reset> .def 0 .latch <Input> <Output>	'sequential' cell in external library Translate to '.latch'

Case Study

- A part of KNICS RPS BP Logic (*th_LO_SG1_LEVEL_Trip*)

Design (FBD)



It consist of 6 shutdown logic (792 FB)
We only used 17 FB

Verilog HDL

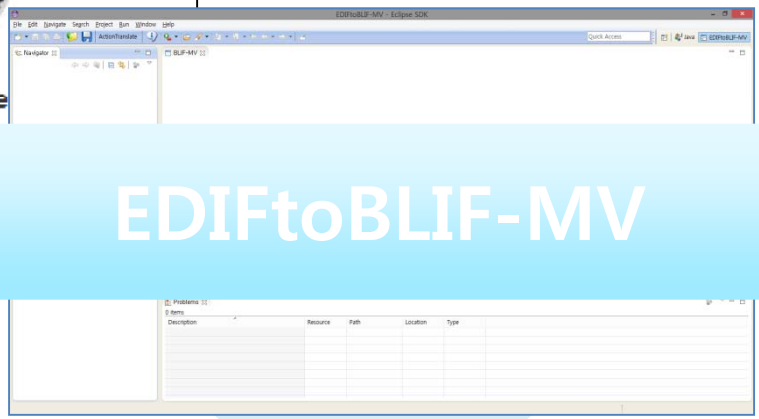
```
module th_LO_SG1_LEVEL_Trip_Logic(clk, reset, f_LO_SG1_LEVEL_Val_Out,
input clk;
input reset;
input [15:0] f_LO_SG1_LEVEL_Val_Out;
output th_LO_SG1_LEVEL_Trip_Logic;
wire cond_A;
wire cond_B;
wire cond_C;
wire cond_D;
wire [15:0] status;
reg [15:0] prev_status;
initial prev_status = 16'b0000000000000000;
cond_A_th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M1(clk,
cond_B_th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M2(clk,
cond_C_th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M3(clk,
cond_D_th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M4(clk,
th_LO_SG1_LEVEL_Trip_Logic_Processing_th_LO_SG1_LEVEL_Trip_Logic th_L
status_th_LO_SG1_LEVEL_Trip_Logic th_LO_SG1_LEVEL_Trip_Logic_M6(clk,
always @(posedge clk) begin
if(reset) begin
prev_status = 16'b0000000000000000;
end else begin
prev_status = status;
end
end
endmodule

module cond_A_th_LO_SG1_LEVEL_Trip_Logic(clk, f_LO_SG1_LEVEL_Val_Out,
// input, output variables
input clk;
input [15:0] f_LO_SG1_LEVEL_Val_Out;
input [15:0] k_LO_SG1_LVL_Trip_Set;
output out;
```

Case Study

Commercial Tool

Libero POWER MATTERS

Microsemi FPGA and SoC Developme



EDIFtoBLIF-MV

Synthesis

EDIF (.edn)

```
(edif th_LO_SG1_LEVEL_Trip_Logic
  (edifVersion 2 0 0)
  (edifLevel 0)
  (keywordMap (keywordLevel 0))
  (status
    (written
      (timeStamp 2013 3 1 20 11 53)
      (author "Synopsys, Inc.")
      (program "Synplify Pro" (version "E-2010.09A-1, mapper mapact,
    )
  )
  (library PA3
    (edifLevel 0)
    (technology (numberDefinition ))
    (cell XA1A (cellType GENERIC)
      (property dont_touch (string "false"))
      (view prim (viewType NETLIST)
        (interface
          (port Y (direction OUTPUT)
          (property function (string "!(A ^ B) & C"))
)
          (port A (direction INPUT)
)
          (port B (direction INPUT)
)
          (port C (direction INPUT)
)
        )
        (property is_combinational (integer 1))
      )
    )
```

Translation

BLIF-MV (.mv)

```
# cell th_LO_SG1_LEVEL_Trip_Logic
.model th_LO_SG1_LEVEL_Trip_Logic
# interface - I/O ports
.inputs reset f_LO_SG1_LEVEL_Val_Out<0> f_LO_SG1_LEVEL_Val_Out<1> f_LO_SG1
.outputs th_LO_SG1_LEVEL_Trip_Logic
.names N_16$M6 N_10$M6 reset_pad prev_status_RNO_0
.def 0
0 0 0 1
0 1 0 1
1 0 0 1
# latches
# init
.r prev_status_0
.def 0
.latch prev_status_RNO_0 prev_status_0
.names N_5$M5 th_LO_SG1_LEVEL_Trip_Logic_pad
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<15> f_LO_SG1_LEVEL_Val_Out_pad_15
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<14> f_LO_SG1_LEVEL_Val_Out_pad_14
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<13> f_LO_SG1_LEVEL_Val_Out_pad_13
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<12> f_LO_SG1_LEVEL_Val_Out_pad_12
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<11> f_LO_SG1_LEVEL_Val_Out_pad_11
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<10> f_LO_SG1_LEVEL_Val_Out_pad_10
.def 0
1 1
```

Case Study

EDIF

```
(edif th_LO_SG1_LEVEL_Trip_Logic
  (edifVersion 2 0 0)
  (edifLevel 0)
  (keywordMap (keywordLevel 0))
  (status
    (written
      (timeStamp 2013 3 1 20 11 53)
      (author "Synopsys, Inc.")
      (program "Synplify Pro" (version "E-2010.09A-1, mapper
     )
   )
  (library PA3
    (edifLevel 0)
    (technology (numberDefinition ))
    (cell XA1A (cellType GENERIC)
      (property dont_touch (string "false"))
      (view prim (viewType NETLIST)
        (interface
          (port Y (direction OUTPUT)
          (property function (string "!(A ^ B) & C"))
)
          (port A (direction INPUT)
)
          (port B (direction INPUT)
)
          (port C (direction INPUT)
)
        )
       (property is_combinational (integer 1))
      )
    )
    (cell VCC (cellType GENERIC)
      (property dont_touch (string "false"))
       (view prim (viewType NETLIST)
         (interface
           (port Y (direction OUTPUT)
```

Translation

BLIF-MV

```
# cell th_LO_SG1_LEVEL_Trip_Logic
.model th_LO_SG1_LEVEL_Trip_Logic
# interface - I/O ports
.inputs reset f_LO_SG1_LEVEL_Val_Out<0> f_LO_SG1_LEVEL_Val_Out<1>
.outputs th_LO_SG1_LEVEL_Trip_Logic
.names N_16$M6 N_10$M6 reset_pad prev_status_RNO_0
.def 0
0 0 0 1
0 1 0 1
1 0 0 1
# latches
# init
.r prev_status_0
.def 0
.latch prev_status_RNO_0 prev_status_0
.names N_5$M5 th_LO_SG1_LEVEL_Trip_Logic_pad
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<15> f_LO_SG1_LEVEL_Val_Out_pad_15
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<14> f_LO_SG1_LEVEL_Val_Out_pad_14
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<13> f_LO_SG1_LEVEL_Val_Out_pad_13
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<12> f_LO_SG1_LEVEL_Val_Out_pad_12
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<11> f_LO_SG1_LEVEL_Val_Out_pad_11
.def 0
1 1
.names f_LO_SG1_LEVEL_Val_Out<10> f_LO_SG1_LEVEL_Val_Out_pad_10
.def 0
1 1
```

	Verilog HDL	EDIF	BLIF-MV from Verilog	BLIF-MV from EDIF
LoC	288	1210	4944	344
Node	8 modules	2 libraries 7 cells	24 models	7 models

Case Study

Results of Equivalence Checking with VIS

```
vis> read_blif_mv from_edif.mv
Warning: Some variables are unused in model th_LO_SG1_LEVEL_Trip_Logic.
vis> flatten_hierarchy
vis> seq_verify from_verilog.mv
Networks are sequentially equivalent.

vis>
```

```
assign SEL_BOOL_35 = SEL_BOOL_def35(
  cond_D_th_LO_SG1_LEVEL_Trip_Logic,
  1'b0,
  th_prev_LO_SG1_LEVEL_Trip_Logic);
```

Fault injection

```
assign SEL_BOOL_35 = SEL_BOOL_def35(
  cond_D_th_LO_SG1_LEVEL_Trip_Logic,
  th_prev_LO_SG1_LEVEL_Trip_Logic,
  1'b0);
```

```
vis> read_blif_mv from_edif.mv
Warning: Some variables are unused in model th_LO_SG1_LEVEL_Trip_Logic.
vis> flatten_hierarchy
vis> seq_verify from_verilog_fault.mv
--State 0:
prev_status<0>:0
prev_status<10>:0
prev_status<11>:0
prev_status<12>:0
prev_status<13>:0
prev_status<14>:0
prev_status<15>:0
prev_status<1>:0
prev_status<2>:0
prev_status<3>:0
prev_status<4>:0
prev_status<5>:0
prev_status<6>:0
prev_status<7>:0
prev_status<8>:0
prev_status<9>:0
prev_status_0$NTK2:0
                                        ...
--Goes to state 1:
prev_status_0$NTK2:1
th_LO_SG1_LEVEL_Trip_Logic_M6.status_0$NTK2:1
--On input:
f_LO_SG1_LEVEL_Val_Out<0>:1
f_LO_SG1_LEVEL_Val_Out<10>:0
f_LO_SG1_LEVEL_Val_Out<11>:0
f_LO_SG1_LEVEL_Val_Out<12>:0
f_LO_SG1_LEVEL_Val_Out<13>:0
f_LO_SG1_LEVEL_Val_Out<14>:0
f_LO_SG1_LEVEL_Val_Out<15>:0
f_LO_SG1_LEVEL_Val_Out<1>:0
f_LO_SG1_LEVEL_Val_Out<2>:1
f_LO_SG1_LEVEL_Val_Out<3>:1
f_LO_SG1_LEVEL_Val_Out<4>:0
f_LO_SG1_LEVEL_Val_Out<5>:1
f_LO_SG1_LEVEL_Val_Out<6>:0
f_LO_SG1_LEVEL_Val_Out<7>:0
f_LO_SG1_LEVEL_Val_Out<8>:0
f_LO_SG1_LEVEL_Val_Out<9>:0
reset:0

Networks are NOT sequentially equivalent.

vis>
```

Conclusion and Future Work

- Confirmation of Correctness of Synthesis
  - Formal Method based Technique (Equivalence Checking)
- Automatic Program Translation
  - Translate into BLIF-MV from EDIF
    - EDIF: Gate-level netlist format of EDA tools
    - BLIF-MV: Front-end format of VIS
- Future Work
  - Formalize the translation rule
  - A whole set of KNICS RPS BP Logic
  - Equivalence Checking between Gate-level Design and Physical Layout

End