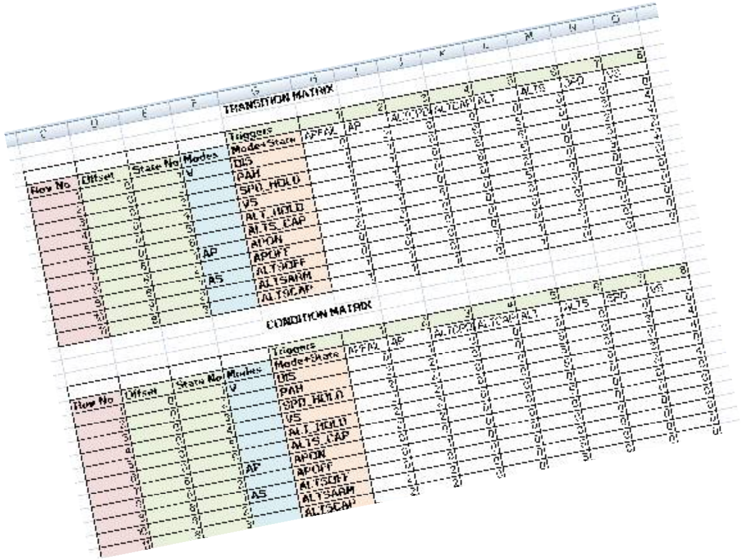


TRANSITION MATRIX

Row No | Offset | State No | Modes | Triggers | Mode+State | APPAL | AP | ALTOPD | ALTCAP | ALT | ALTS | SPD | VS

V: DIS, PAH, SPD_HOLD, VS, ALT_HOLD, ALTS_CAP

AP: APON, APOFF

AS: ALTSOFF, ALTSARM, ALTSCAP

CONDITION MATRIX

Row No | Offset | State No | Modes | Triggers | Mode+State | APPAL | AP | ALTOPD | ALTCAP | ALT | ALTS | SPD | VS

V: DIS, PAH, SPD_HOLD, VS, ALT_HOLD, ALTS_CAP

AP: APON, APOFF

AS: ALTSOFF, ALTSARM, ALTSCAP

Exploring Simulink Design Verifier – 3

Mode Transition Table to Matlab and NuSMV Code

Natasha Y Jeppu

NITK, Surathkal

NuSMV

- This is a symbolic model checker available at http://nusmv.fbk.eu/NuSMV/
- NuSMV is a software tool for the formal verification of finite state systems. It has been developed jointly by FBK-IRST and by Carnegie Mellon University.
- NuSMV language is designed to allow the description of finite state systems that range from completely synchronous to completely asynchronous

Simulink Design Verifier

- This is a tool available from Mathworks
- This tool has been used for demonstrating the correct functionality of safety critical control system elemental blocks and an autopilot mode transition earlier
- https://www.mathworks.com/matlabcentral/fileexchange/51567-exploring-simulink-design-verifier-2
- https://www.mathworks.com/matlabcentral/fileexchange/48858-exploring-design-verifier

Mode Transition Table

- Mode Transition table for an autopilot is described in the earlier examples
- In these files the Mode Transition Table is generalised and a Matlab code is provided that can convert the mode transition table to a Matlab script that can be used by SDV and a NuSMV code that can be used by the NuSMV tool
- Two examples are provided that can help the user to understand the Mode Transition Logic and its implementation
- An excel file is provided to help design the mode transition logic

TRANSITION MATRIX

				Triggers	1	2	3	4	5	6	7	8
Row No	**Offset**	**State No**	**Modes**	**Mode+State**	APFAIL	AP	ALTCPDN	ALTCAP	ALT	ALTS	SPD	VS
1	0	1	**V**	**DIS**	0	2	0	0	0	0	0	0
2	0	2		**PAH**	1	1	0	6	5	0	3	4
3	0	3		**SPD_HOLD**	1	1	0	6	5	0	2	4
4	0	4		**VS**	1	1	0	6	5	0	3	2
5	0	5		**ALT_HOLD**	1	1	0	0	2	0	3	4
6	0	6		**ALTS_CAP**	1	1	5	0	5	0	0	0
7	6	1	**AP**	**APON**	2	2	0	0	0	0	0	0
8	6	2		**APOFF**	0	1	0	0	0	0	0	0
9	8	1	**AS**	**ALTSOFF**	0	0	0	0	0	2	0	0
10	8	2		**ALTSARM**	1	1	0	3	1	1	0	0
11	8	3		**ALTSCAP**	1	1	1	0	1	0	0	0

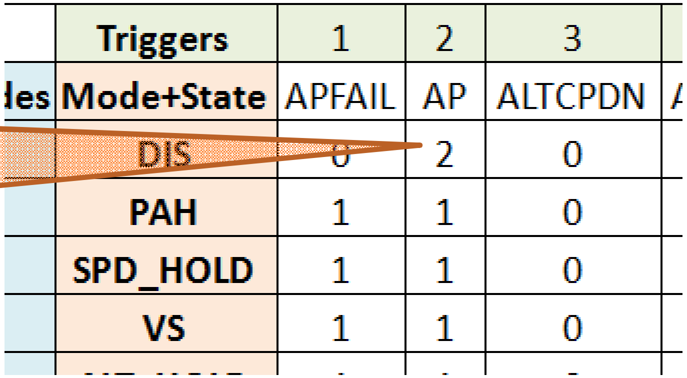
TRANSITION MATRIX

- The first col gives the row number. This defines the number of rows in the TM
- The second col defines the offset. This data is required by the tool. This defines the row number to be added to the state number to arrive at the actual row in the TM matrix
- In the table there are 3 modes defined in col 4. Each mode can exist independently
- Each mode can exist in only one state (col 3)
- Col 5 gives the names of the states which can be used for comments or as variables in NuSMV code

TRANSITION MATRIX

- The second row has the trigger names. Triggers are defined from left to right in the order of decreasing priority
- The mode transition is triggered by a single trigger at a time. In case of multiple triggers the trigger with the highest priority is selected
- The rest for the table data defines the transition

When Trigger AP happens the MTL changes to PAH (2) if in DIS (1)



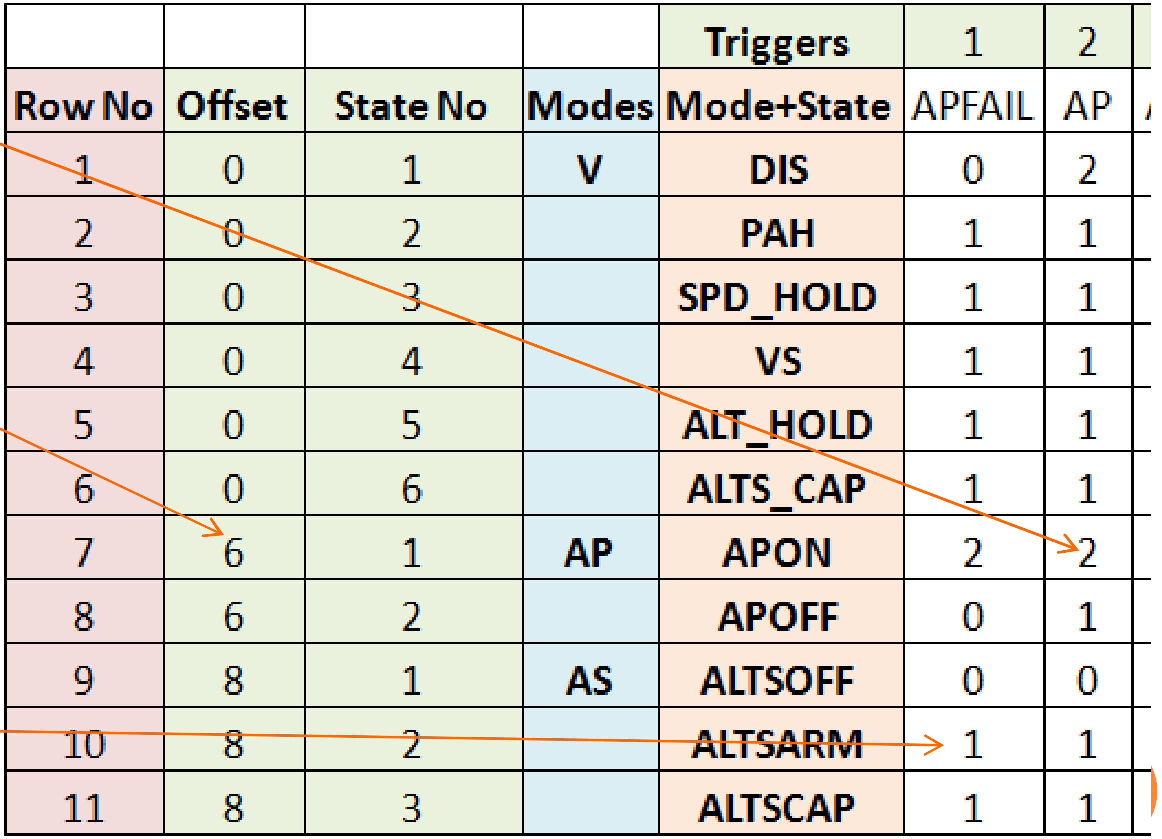
	Triggers	1	2	3
les	Mode+State	APFAIL	AP	ALTCPDN
	DIS	0	2	0
	PAH	1	1	0
	SPD_HOLD	1	1	0
	VS	1	1	0

TRANSITION MATRIX

If in APON (1 of AP mode) an AP trigger happens transit to APOFF (2 of AP mode)

The offset 6 is added to APON (1) to arrive at row number 7 of the table.

If in ALTSARM (2 of AS mode) and trigger APFAIL happens transit to ALTSOFF (1 of AS mode)



				Triggers	1	2
Row No	Offset	State No	Modes	Mode+State	APFAIL	AP
1	0	1	V	DIS	0	2
2	0	2		PAH	1	1
3	0	3		SPD_HOLD	1	1
4	0	4		VS	1	1
5	0	5		ALT_HOLD	1	1
6	0	6		ALTS_CAP	1	1
7	6	1	AP	APON	2	2
8	6	2		APOFF	0	1
9	8	1	AS	ALTSOFF	0	0
10	8	2		ALTSARM	1	1
11	8	3		ALTSCAP	1	1

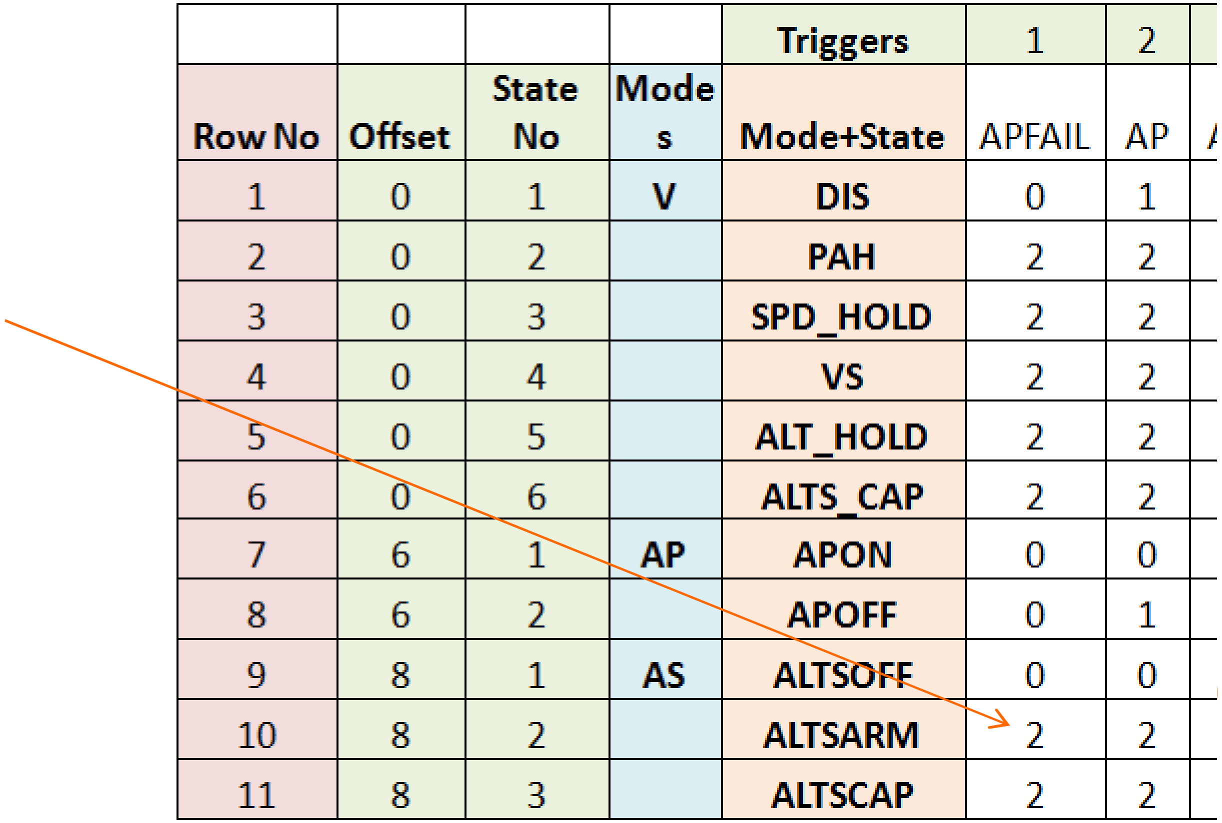
CONDITION MATRIX

				Triggers	1	2	3	4	5	6	7	8
Row No	**Offset**	**State No**	**Modes**	**Mode+State**	APFAIL	AP	ALTCPDN	ALTCAP	ALT	ALTS	SPD	VS
1	0	1	**V**	**DIS**	0	1	0	0	0	0	0	0
2	0	2		**PAH**	2	2	0	0	5	0	3	4
3	0	3		**SPD_HOLD**	2	2	0	0	5	0	0	4
4	0	4		**VS**	2	2	0	0	5	0	3	0
5	0	5		**ALT_HOLD**	2	2	0	0	0	0	3	4
6	0	6		**ALTS_CAP**	2	2	0	0	5	0	0	0
7	6	1	**AP**	**APON**	0	0	0	0	0	0	0	0
8	6	2		**APOFF**	0	1	0	0	0	0	0	0
9	8	1	**AS**	**ALTSOFF**	0	0	0	0	0	0	0	0
10	8	2		**ALTSARM**	2	2	0	0	5	0	0	0
11	8	3		**ALTSCAP**	2	2	0	0	5	0	0	0

Condition Matrix

- The transition happens if the corresponding guard conditions are TRUE. This is given in the condition table

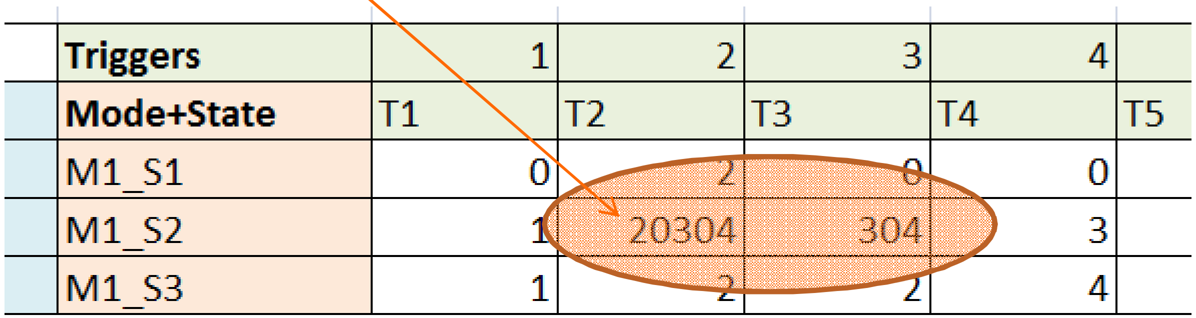
If in ALTSARM (2 of AS mode) and trigger APFAIL happens transit to ALTSOFF (1 of AS mode) **IF Condition C2 is TRUE**



				Triggers	1	2
Row No	Offset	State No	Modes	Mode+State	APFAIL	AP
1	0	1	V	DIS	0	1
2	0	2		PAH	2	2
3	0	3		SPD_HOLD	2	2
4	0	4		VS	2	2
5	0	5		ALT_HOLD	2	2
6	0	6		ALTS_CAP	2	2
7	6	1	AP	APON	0	0
8	6	2		APOFF	0	1
9	8	1	AS	ALTSOFF	0	0
10	8	2		ALTSARM	2	2
11	8	3		ALTSCAP	2	2

Condition Matrix

- It is a good practice to use Condition 1 as always TRUE
- Use 0 in Transition and Condition table if there is no effect of the trigger
- Multiple transitions can be defined from a state. At present three transitions can be defined. Priority is right to left. In 020304, 04 has higher priority than 03 and 02



Triggers	1	2	3	4	
Mode+State	T1	T2	T3	T4	T5
M1_S1	0	2	0	0	
M1_S2	1	20304	304	3	
M1_S3	1	2	2	4	

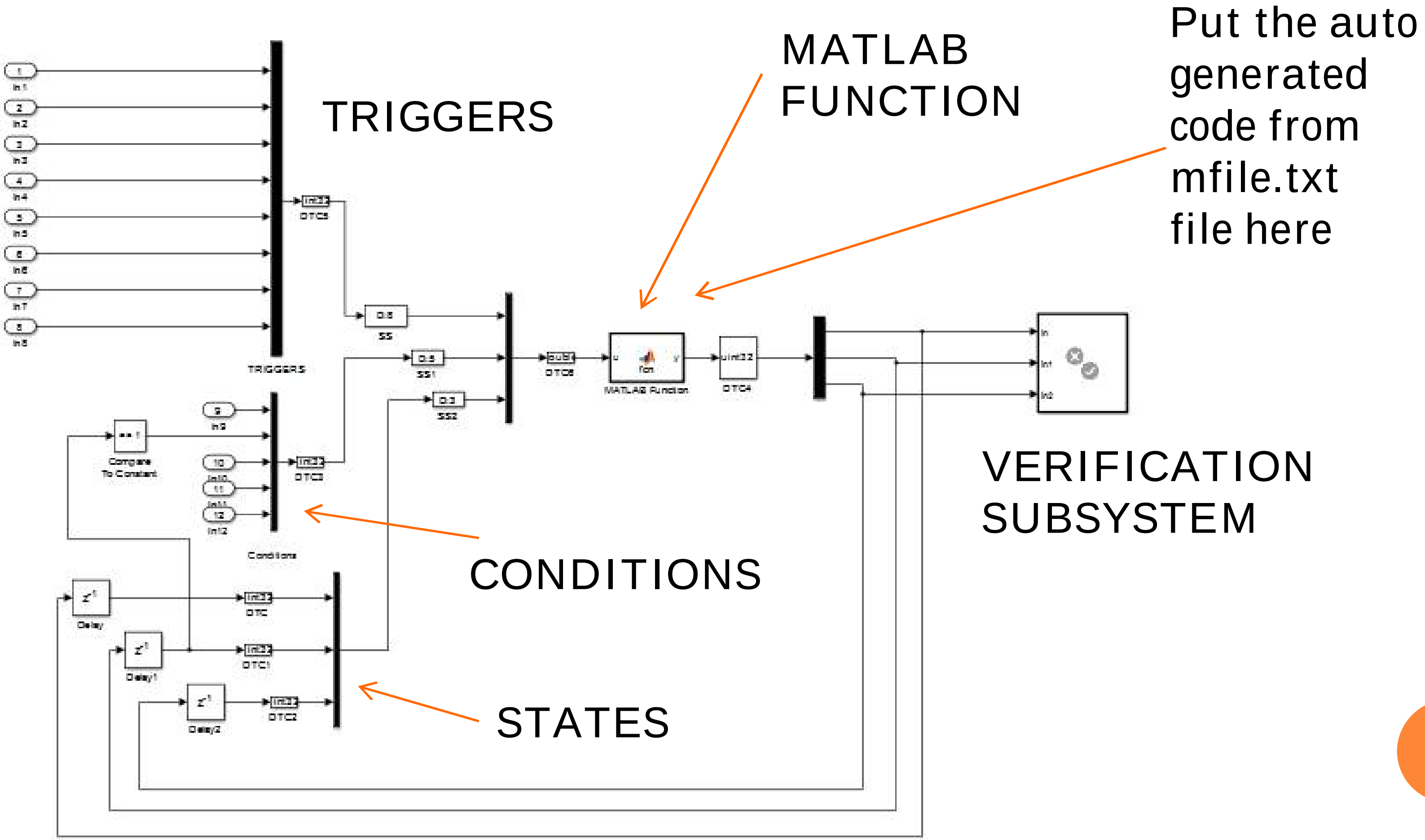
MATLAB SCRIPTS

- Two matlab scripts are provided to generate the Matlab function for SDV and the NuSMV code.
- make_mtl_code_M_gen.m
- make_mtl_code_SMV_gen.m

- Design the mode transition for your system in the excel file
- Copy the data into the variables in the script file from the excel file
- Execute the files to generate the code

USING SDV

- An example model is provided.



TRIGGERS

MATLAB FUNCTION

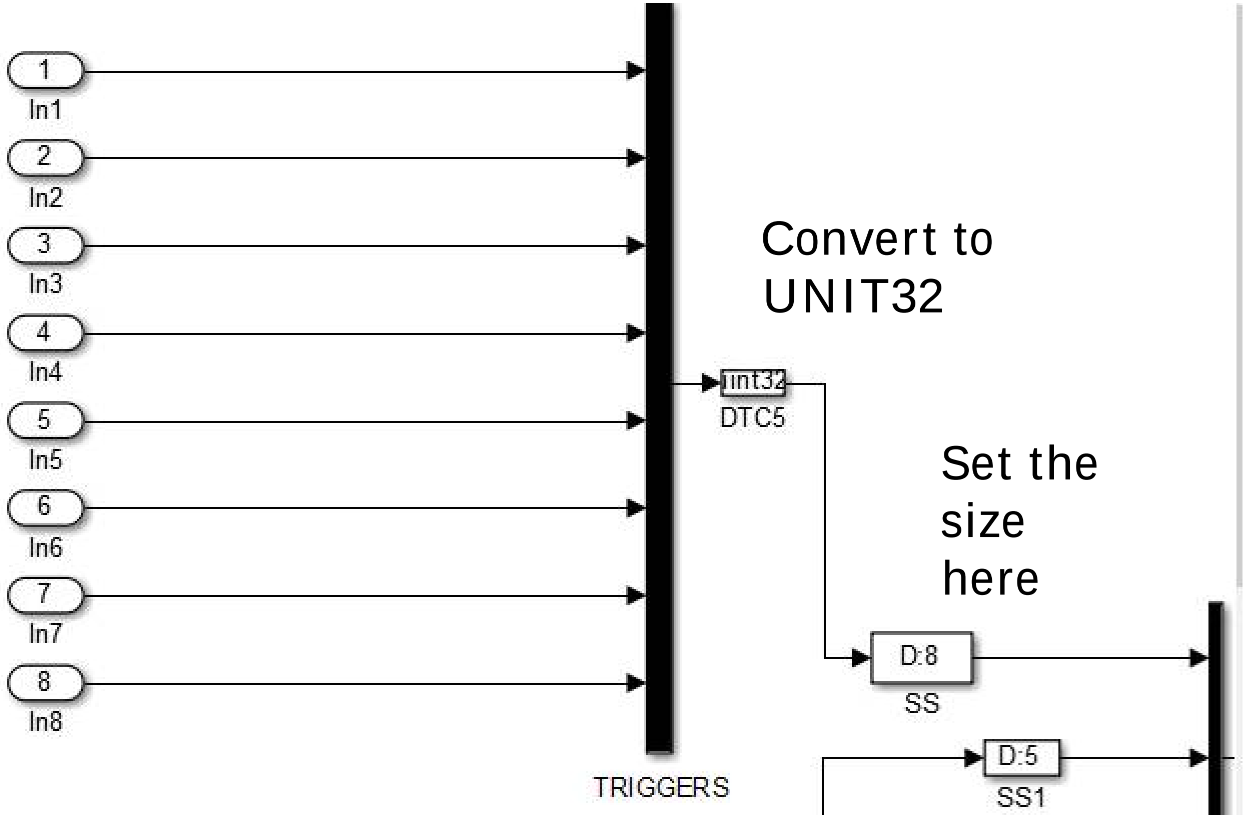
Put the auto generated code from mfile.txt file here

VERIFICATION SUBSYSTEM

CONDITIONS

STATES

Using SDV

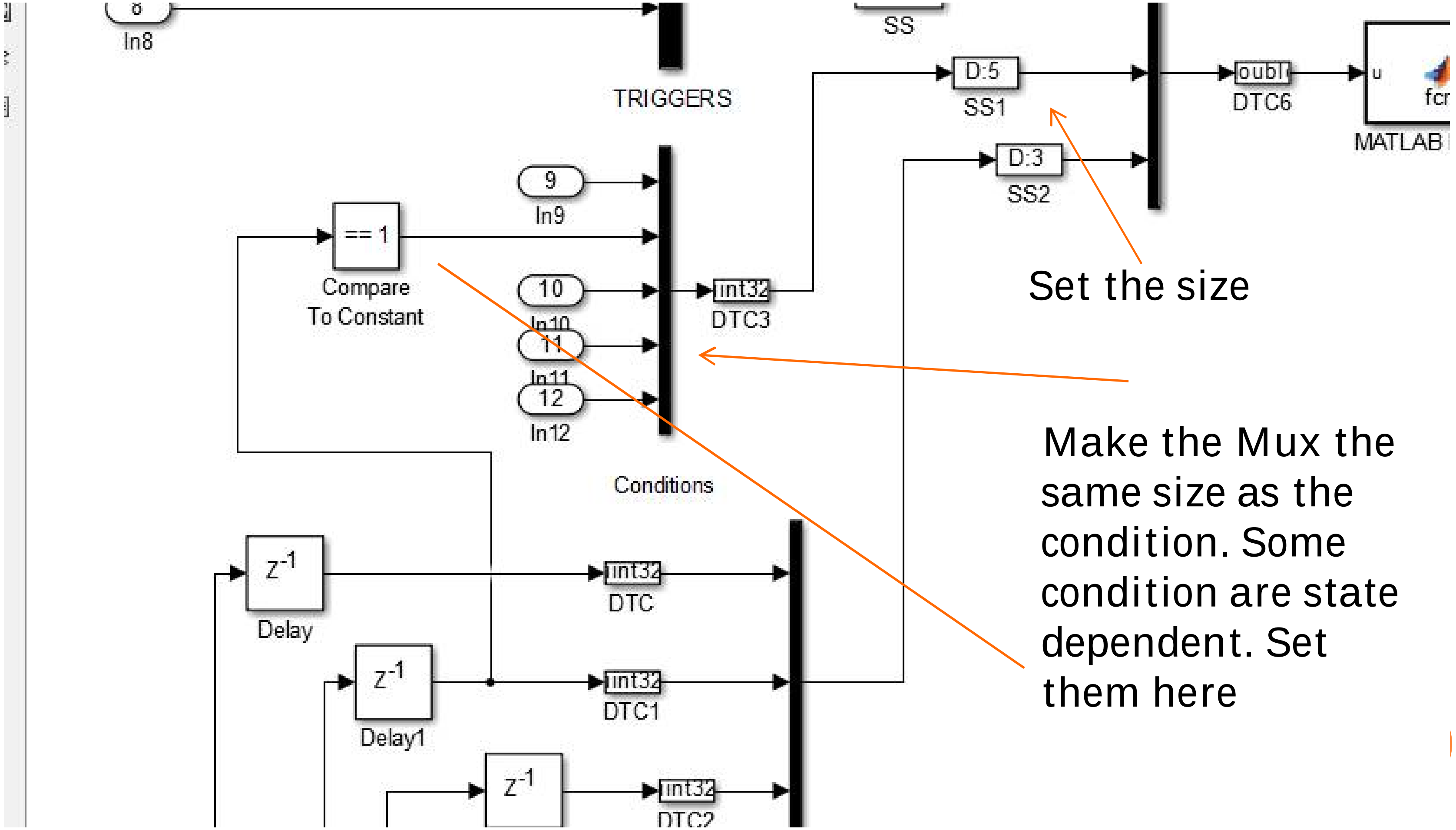


Convert to UNIT32

Set the size here

Set the number of triggers to the number of triggers in your MTL. All inputs are Boolean

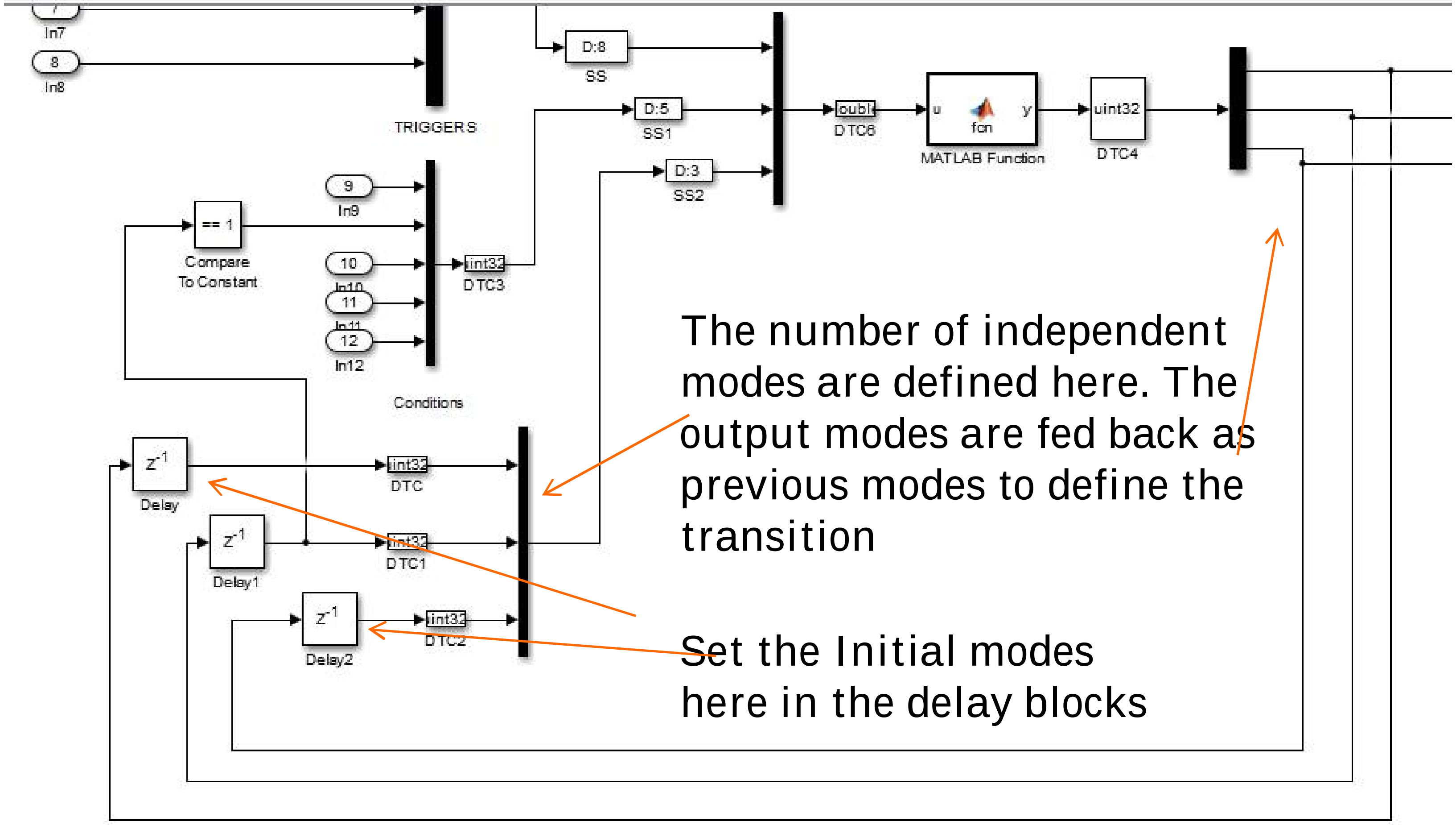
Using SDV



Set the size

Make the Mux the same size as the condition. Some condition are state dependent. Set them here

USING SDV

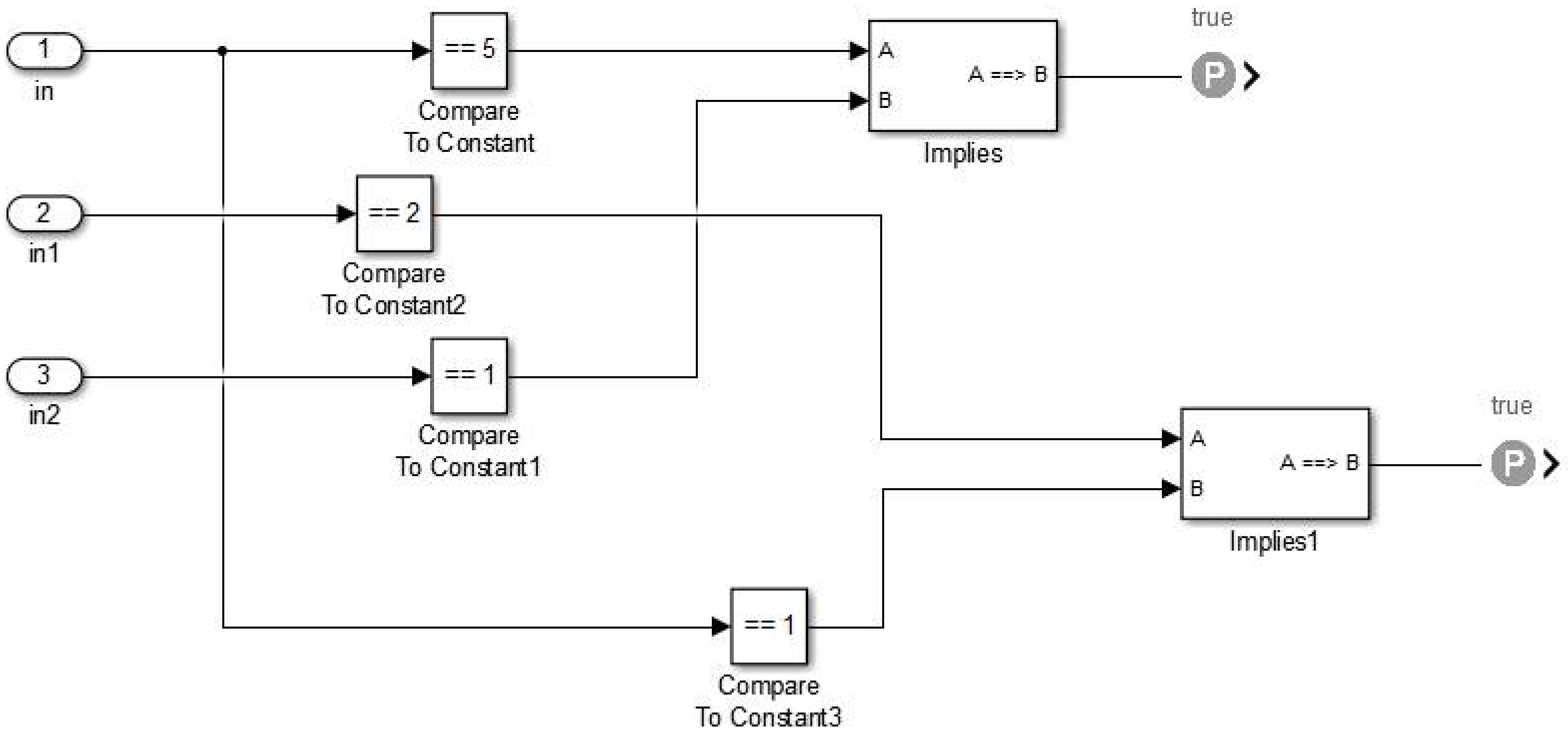


The number of independent modes are defined here. The output modes are fed back as previous modes to define the transition

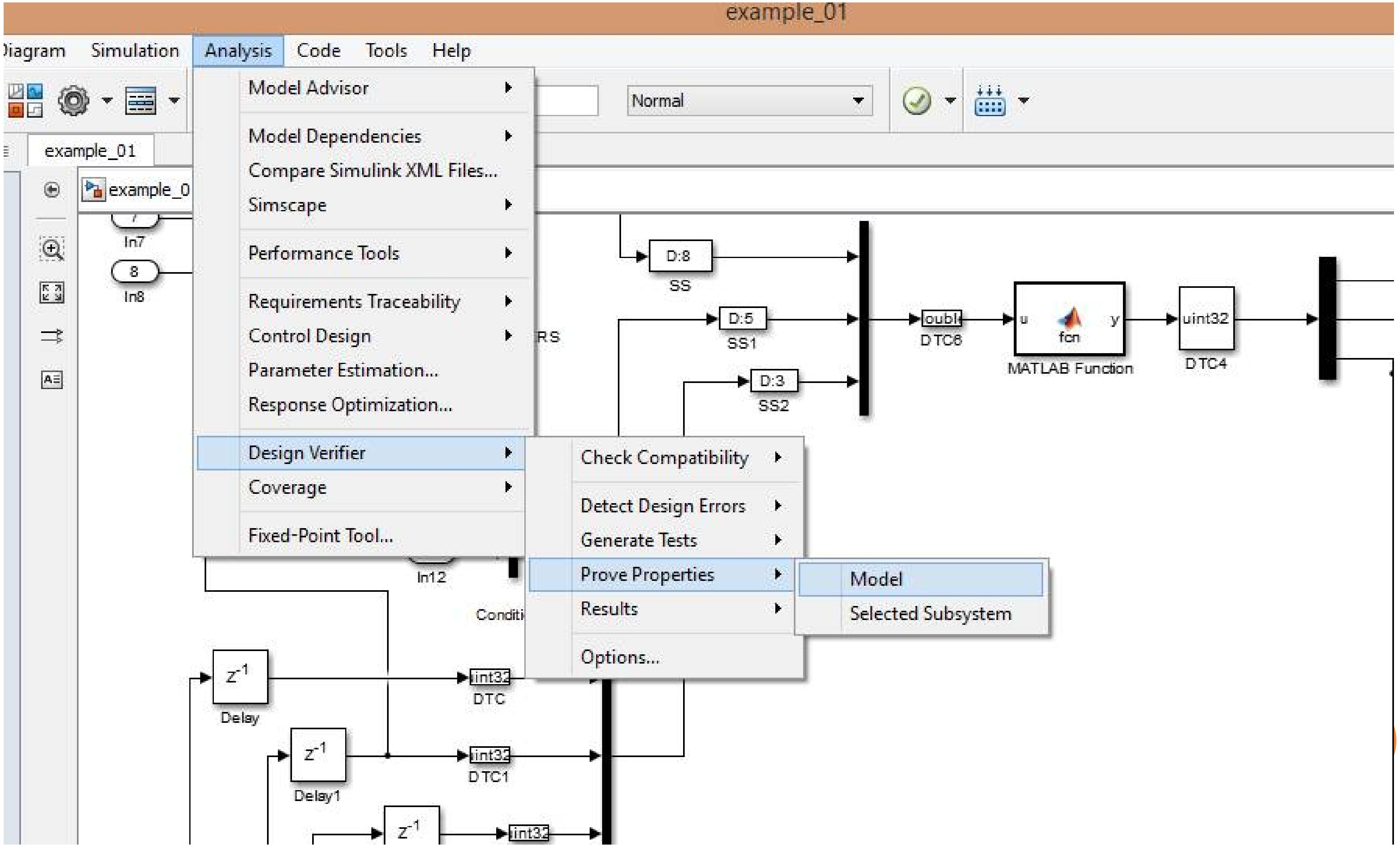
Set the Initial modes here in the delay blocks

USING SDV

- Make the LTL assertions in the verification subsystem

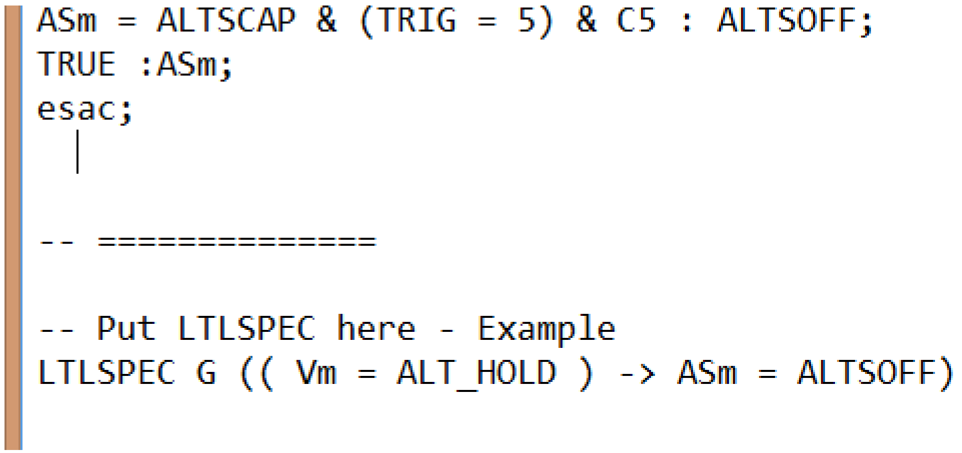


PROVE PROPERTIES



USING NUSMV

- The NuSMV code is given is test.smv file. Rename this and copy it to the NuSMV directory
- The NuSMV software can be downloaded from the internet
- You will have to put in the LTL in the end of the file



```
ASm = ALTSCAP & (TRIG = 5) & C5 : ALTSOFF;
TRUE :ASm;
esac;


-- ==============


-- Put LTLSPEC here - Example
LTLSPEC G (( Vm = ALT_HOLD ) -> ASm = ALTSOFF)
```

LTL Specifications

SIMULATION RESULTS

- Two examples are taken to compare NuSMV and SDV performance and behaviour
- The code for Matab and NuSMV are very similar.
- The LTL properties are very easy to describe in both SDV and NuSMV code
- The comments provided with the code help review the MTL easily
- Both provide the counter examples if the assertion is proved false. Both SDV and NuSMV have similar counter examples.
- SDV is very slow compared to NuSMV

THANK YOU

- Please feel free to email me in case you find any issues or require any clarification
- Natasha.jeppu@gmail.com