SPECIAL BIT PATTERN INJECTION IN SIMULATION VERIFICATION BY LEVERAGING FORMAL VERIFICATION

Viraj Y. Rawal Vishwajith V. Rao Nivin N. George *

*Synopsys India Private Limited, Bengaluru, Karnataka, India

Motivation

Problem: Low Simulation Coverage for Special Bit Pattern (SBP) in data structure

- Packet-based communication protocols have well-defined data packet structures

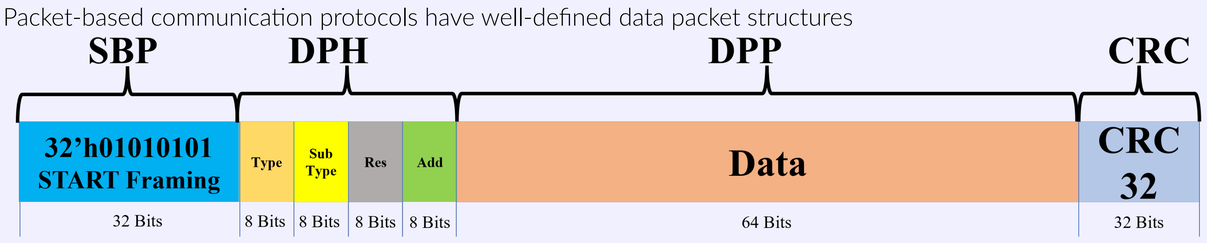


Figure 1. Data Packet Structure

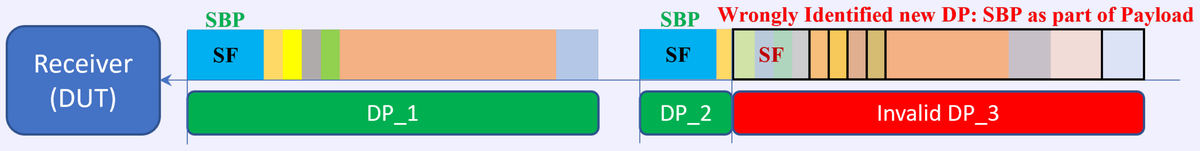


Figure 2. Wrongly Identified Data Packet within Data Packet due to Special Bit Pattern (SBP) inside Payload structure

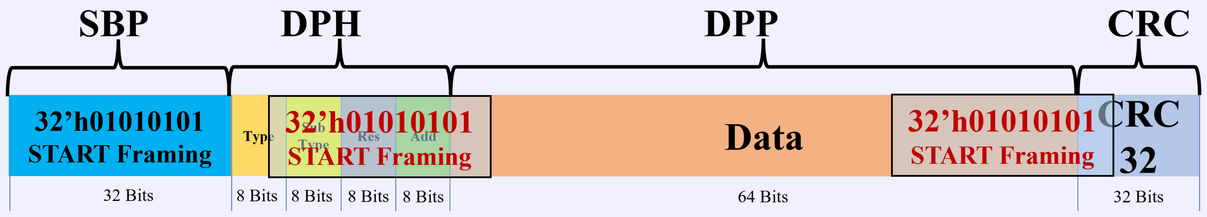


Figure 3. USB 32 Data packet framing injected with the Special Bit Pattern (SBP)

Challenges:

- Random or Directed Simulation is not exhaustive and it is not possible to generate all possible data combinations
- Difficult to generate required stimulus for specific scenarios

Main Idea(Overview)

Formal Verification Environment (FVE):

1. **Constraints:** System Verilog Assume properties are deployed to constrain the Data Packet Header Fields to follow the Design Specification.
2. **Stimulus Generation:** System Verilog Cover properties help to generate the required stimulus for injecting Special Bit Patterns (SBP) anywhere in the Data Packet Structure including DPH and CRC32 locations in the packet.
3. **Exclusion Generation:** Helps to exclude the scenarios which are uncoverable – minimize Verification Engineer's efforts

Simulation Verification Environment (SVE):

1. **Validation:** System Verilog Assume properties used in FVE for DPH fields, is used as System Verilog Assertions in a Simulation Verification Environment (SVE) for validation of no over-constrains in FVE.
2. **Improvement of Coverage:** Hard-to-cover scenarios can be covered using stimulus generation – improve Code Coverage, Functional Coverage – Cover Points, Cover Groups

	SBP	8'h01	8'h01	8'h01	8'h01		SBP	8'h01	8'h01	8'h01	8'h01
		=	=	=	=			=	=	=	=
Byte Positions	0	1	2	3	4	…	11	12	13	14	15
DWORD	Type	Sub Type	Res	Address		Data		CRC32			

```
for (i=0; i< 13; i=i+1)   Pattern Match[i] = ((BYTE[i]==SBP[0]) && (BYTE[i+1]==SBP[1]) &&
                                              (BYTE[i+2]==SBP[2]) && (BYTE[i+3]==SBP[3]) )
cover_match_1_loc: cover property (@(posedge clk) (Pattern_Match[1]==1'b1));
cover_match_12_loc: cover property (@(posedge clk) (Pattern_Match[12]==1'b1));
```

Figure 4. Cover Property for Special Bit Pattern (SBP) Detection in Data Packet Frame

Results and discussion

Results

Table 1. Results of cover properties targeting scenarios of interest in Formal Verification Environment (FVE) and Simulation Verification Environment (SVE)

Pattern	Formal Verification Environment (FVE)		Simulation Verification Environment (SVE)	
	Covered	Uncoverable	Hits	Excluded
SBP1	24	49	24	49
SBP2	23	45	23	45
SBP3	19	77	19	77
Total	66	171	66	171

1. Stimulus generated from 66 cover properties witness traces in Formal Verification Environment (FVE) were utilized to cover the Special Bit Pattern (SBP) injection in Data Packet
2. These stimuli in the Simulation Verification Environment (SVE) were augmented by extra stimuli that are the input from the Formal Verification Environment (FVE) and 100% scenarios were found to be covered in the VCS-based Simulation Verification Environment (SVE)
3. As seen from Table 1, Formal Verification can assist in generating stimuli of interest in the Simulation Verification Environment (SVE)
4. Here, 171 Cover properties are excluded from the Simulation Verification Environment (SVE) since they are uncoverable in Formal Verification Environment (FVE)

Summary

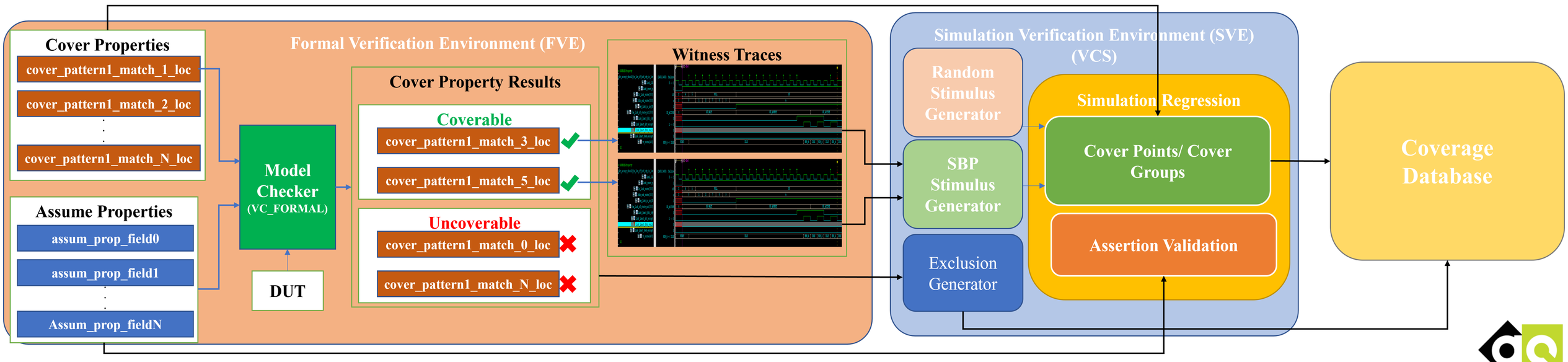
1. System Verilog Assume Properties are used to constrain Data packet fields values to follow the Design Specification
2. System Verilog Cover properties can be used for Special Bit Pattern (SBP) injected at various positions in the Data Packet Structure
3. System Verilog assume properties and System Verilog Cover properties can be used in Simulation Environment (SVE) and Formal Verification environment (FVE)
4. Required Stimulus for Special Bit Pattern (SBP) detection in Data Packets can be generated by using System Verilog cover properties in Formal Verification Environment (FVE)
5. Simulation Coverage (Functional Coverage and Code Coverage) can be enhanced by utilizing these generated stimuli

What are the Main Contributions?

1. **Coverage:** Improving the Simulation Coverage by providing an automated way for generating required stimuli for hard-to-hit scenarios used in Random/Directed Simulation Verification Environment (SVE)
2. **Exclusions:** Improve Verification Engineer's coverage closure efficiency by excluding impossible-to-hit scenarios identified from uncoverable property(ies) results in Formal Verification Environment (FVE)

What are the Advantages?

1. Simulation Coverage can be improved by reaching more cover points and coverage bins triggered by augmenting stimuli generated by Formal Verification Environment (FVE)
2. Cover properties that are failing in Formal Verification Environment (FVE):
   1. Can be excluded in the Coverage matrix using an exclusion generator to improve the impossible-to-hit scenarios
   2. Wasteful efforts for verification engineers can be eliminated by relying on a guarantee for Uncovered scenarios in Formal Verification Environment (FVE)

Practical implications

1. What is the expected run time in Formal Verification if we have 1000+ cover properties?: Properties are not Complex in nature - Run time is within a few minutes - whereas Variable efforts to generate the same stimuli manually by the Verification Engineers.
2. Is pre-requisite knowledge of Formal Verification required to use this utility?: Formal Verification background is not needed since syntax to write System Verilog properties is simple