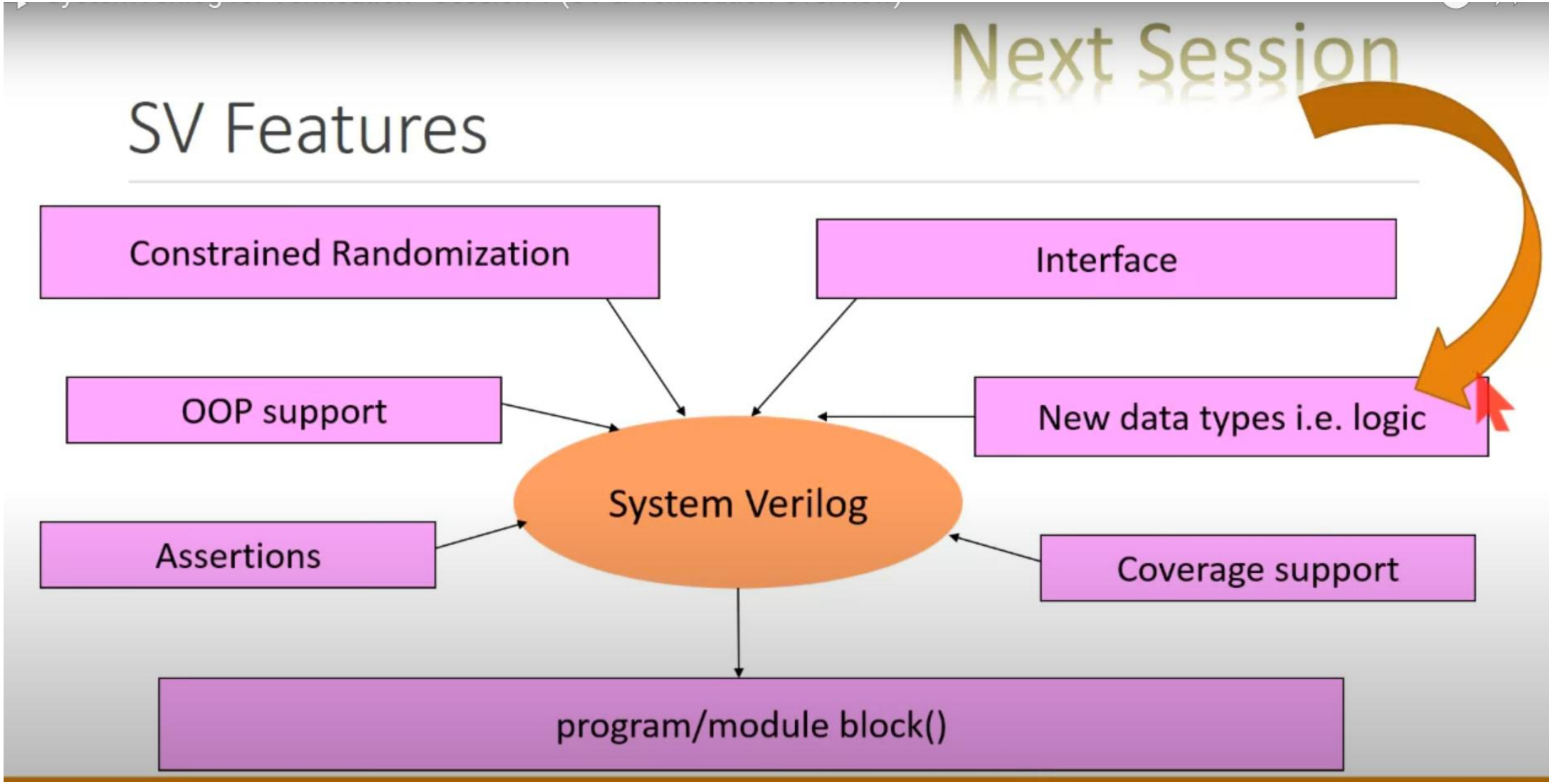

Next Session
SV Features
Constrained Randomization
Interface
OOP support
New data types i.e. logic
System Verilog
Assertions
Coverage support
program/module block()

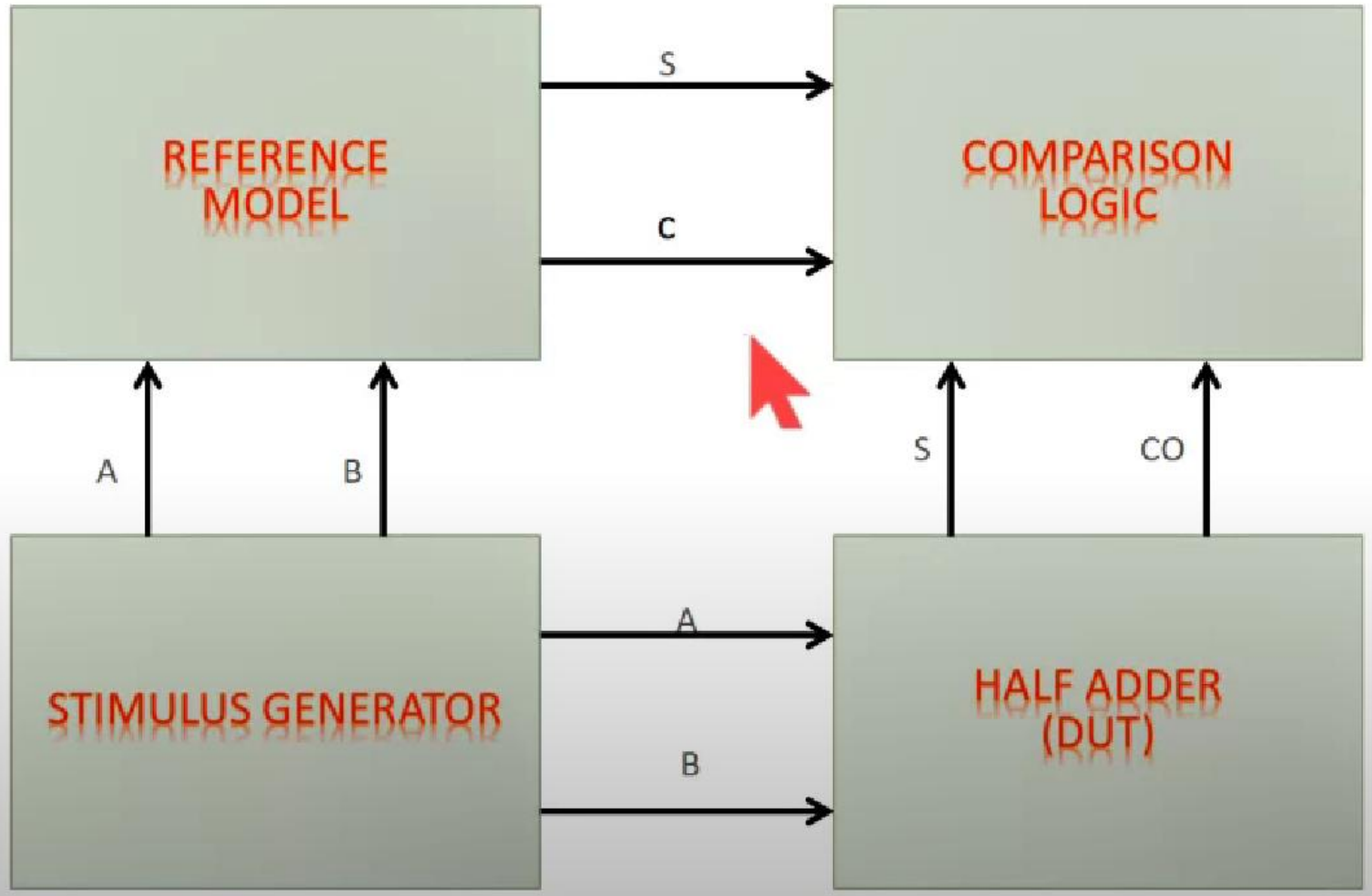
Why SystemVerilog for Verification ?

- Verilog was initially used for writing testbench.
- But, writing complex testbenches is much more of a programming task than describing hardware. No need to synthesize testbench.

- Fact: UVM is replacing SV based verification in industry.
- Still, UVM = Structured SV. Knowing SV based verification helps understanding UVM based verification, else UVM feels like set of magic macros.

UVM = Universal Verification Methodology

How To Verify ?



What is Verification ?

- Complete validation of design functionality
- Make sure design is bug free by verifying it in all aspects ; all possible scenarios

```
module half_adder(s,co,a,b);
  output s,co;
  input a,b;
  xor1 u1(s,a,b);
  nand1 u2 (co,a,b);
endmodule
```

Will this work??

How to make sure design is

bug free??

- Stimulating DUT (Design under test) with all possible scenarios and comparing its output with golden results

What is Verification ?

- Complete validation of design functionality
- Make sure design is bug free by verifying it in all aspects ; all possible scenarios

```
module half_adder(s,co,a,b);
  output s,co;
  input a,b;
  xor1 u1(s,a,b);
  nand1 u2 (co,a,b);
endmodule
```

- Stimulating DUT (Design under test) with all possible scenarios and comparing its output with golden results

What is SystemVerilog ?

- HDVL (Hardware Description and Verification Language)
- Unified design, modeling, verification
- Latest IEEE standard 1800-2012

Agenda

- ✓ What is SystemVerilog ?
- ✓ What is Verification ?
- ✓ How To Verify ?
- ✓ Why SystemVerilog for Verification ?
- ✓ SV Features

SystemVerilog for Verification

SV & VERIFICATION OVERVIEW