

www.verificationexcellence.in

**Verification**  
*Excellence*

# TRANSACTION LEVEL MODELLING

1

**INTRODUCTI  
ON TO OVM &  
UVM**



**6/4/2020**

# TRANSACTION

- **A CLASS OBJECT THAT INCLUDES NECESSARY INFORMATION NEEDED FOR COMMUNICATION BETWEEN TWO COMPONENTS**
- **SIMPLE EXAMPLE CAN BE A READ/WRITE TRANSACTION ON A BUS**
- **OVM/UVM CLASS LIBRARY HAS OVM\_TRANSACTION / UVM\_TRANSACTION DEFINED FOR A BASE TRANSACTION**

# TRANSACTION EXAMPLE

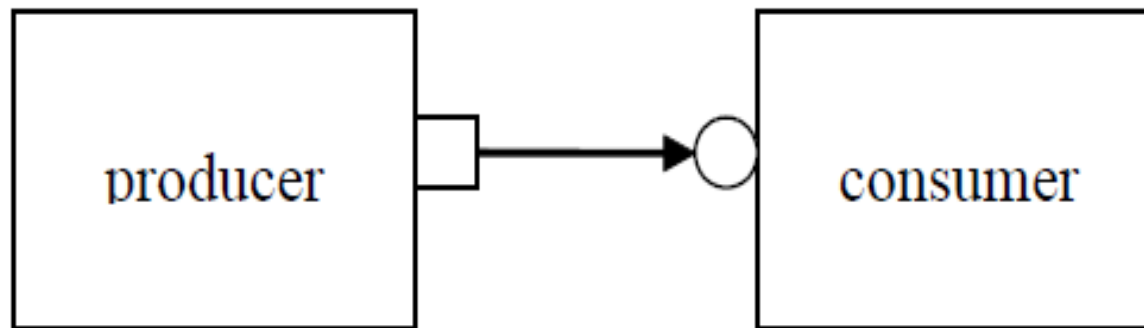
```
class simple_trans extends uvm_transaction;
    rand data_t data;
    rand addr_t addr;
    rand enum {WRITE,READ} kind;
    constraint c1 { addr < 16'h2000; }
    ...
endclass
```

# TRANSACTION LEVEL MODELING (TLM)

- TRANSACTION-LEVEL MODELLING IS AN APPROACH THAT CONSIST OF MULTIPLE PROCESSES COMMUNICATING WITH EACH

OTH

FOR



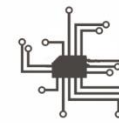
AND

# TRANSACTION LEVEL MODELLING

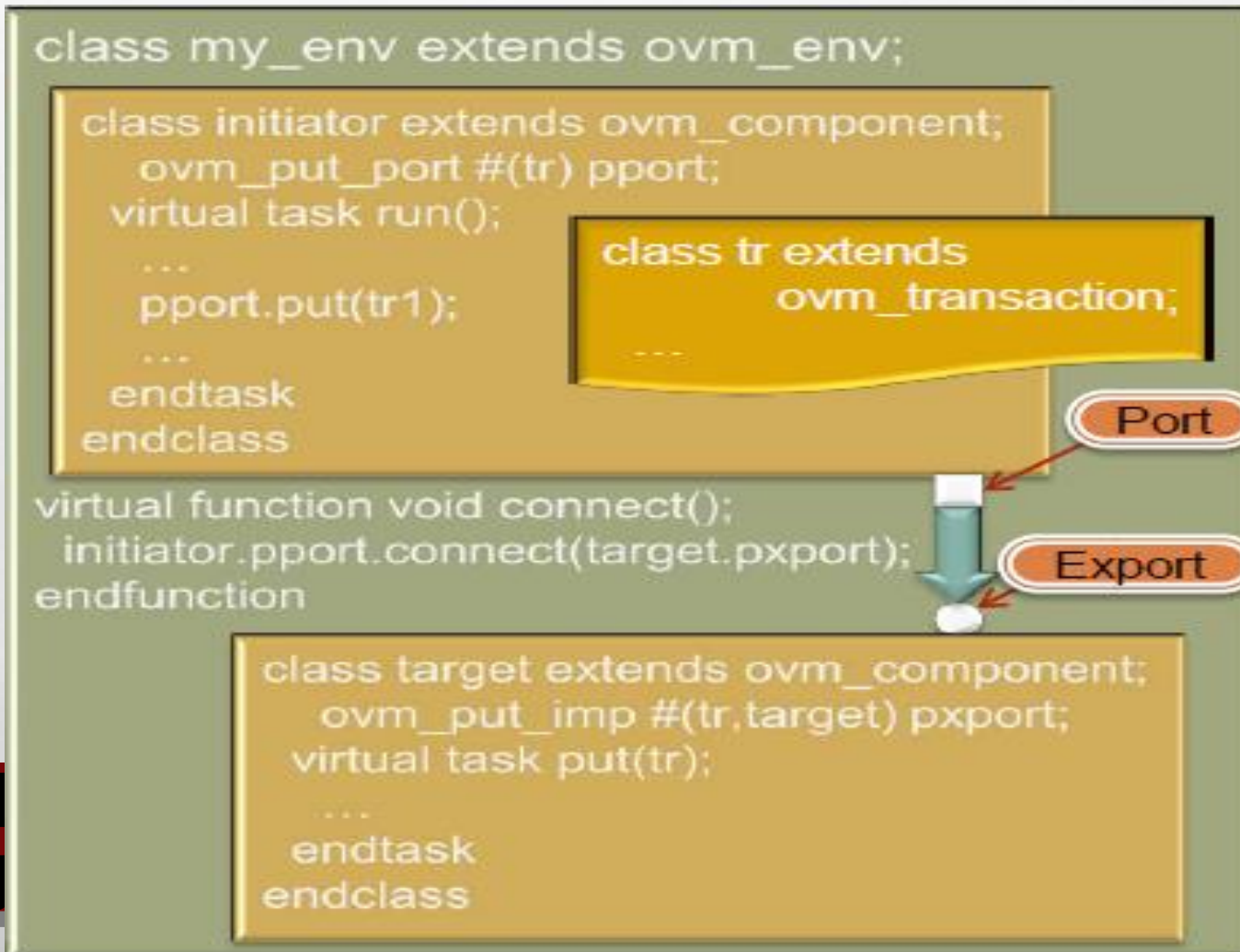
- ABSTRACTS TIME
  - REDUCES THE NUMBER OF ACTIVATION OF PROCESSES
  - RTL EVALUATION OF NETS/SIGNALS VS EVALUATION OF TRANSACTIONS
- ABSTRACTS DATA
  - FORM TRANSACTION OBJECTS INSTEAD OF BIT BY BIT DETAILS
- ABSTRACTS FUNCTION
  - IMPLEMENT FUNCTIONALITY WITH FUNCTIONS/PROCESSES RATHER THAN WITH REAL REGISTERS OR CIRCUITS

# UNDERSTANDING TLM - OVM/UVM

- TLM IS THE BASIS FOR MODULARITY AND RE-USE IN OVM/UVM
- TLM IS ALL ABOUT *COMMUNICATION* THROUGH *METHOD CALLS*
  - A TLM *PORT* SPECIFIES THE “API” TO BE USED
  - A TLM *EXPORT* SUPPLIES THE IMPLEMENTATION OF THE METHODS
- CONNECTIONS ARE BETWEEN PORTS/EXPORTS, NOT COMPONENTS
- TRANSACTIONS ARE *OBJECTS*
- PORTS & EXPORTS ARE PARAMETERIZED BY THE TRANSACTION TYPE BEING COMMUNICATED



# UNDERSTANDING TLM - OVM/UVM





# TLM INTERFACES SUPPORTED

- Unidirectional
  - Blocking (Tasks) – blocks till completion
    - put
    - get/peek
  - Nonblocking (functions)
    - try\_put
    - try\_get/try\_peek
    - write

