

# BEHAVIORAL TEST GENERATION MODELING APPROACH FOR MIXED- SIGNAL IC VERIFICATION

IMSTW2002  
Montreaux 19th June 2002

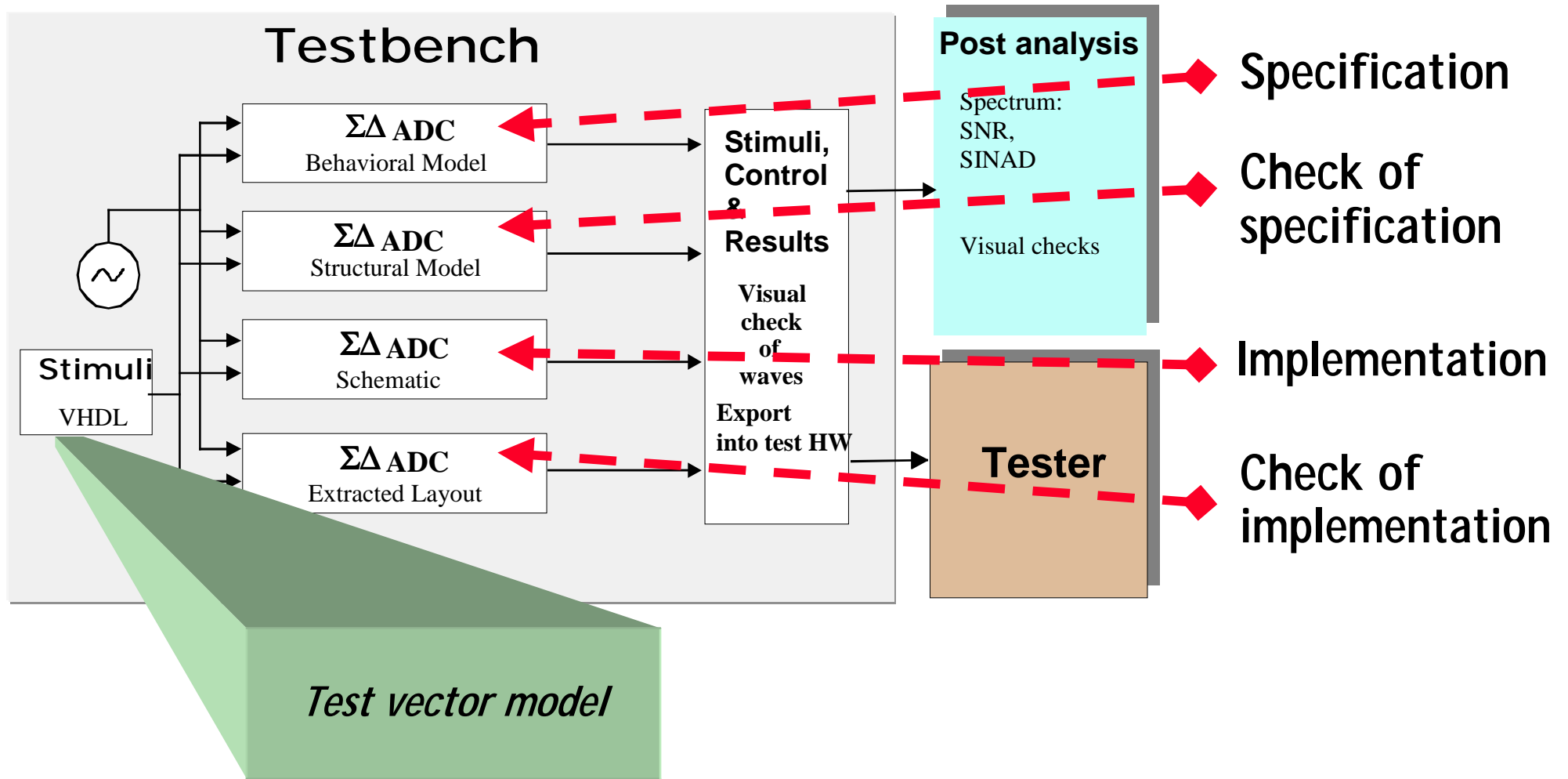
*Veikko Loukusa* Nokia Mobile Phones, Finland

# Contents

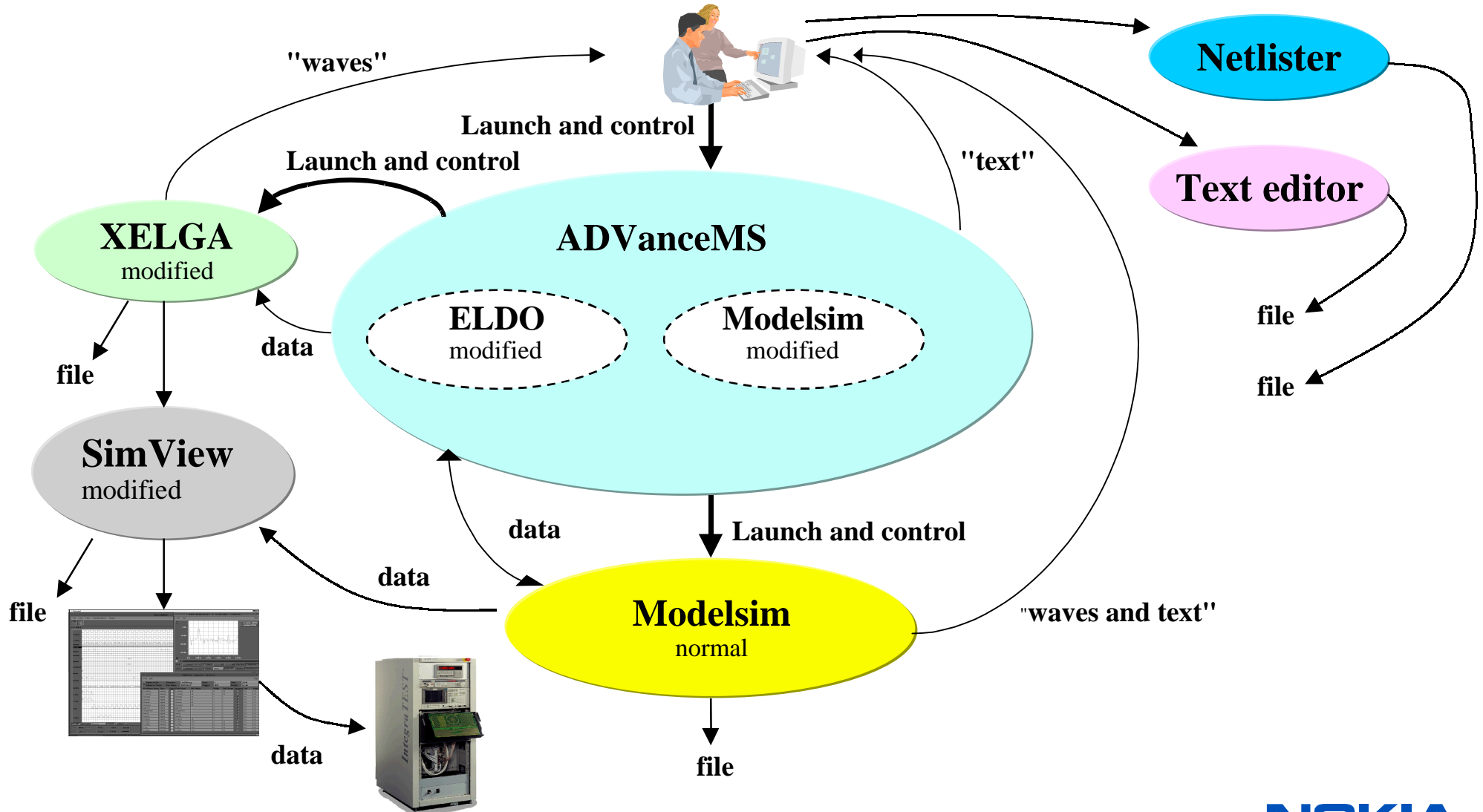
- **Generic Test Generation Modeling**
  - Principle
  - Framework
  - Mechanism
  - Test vector model
- **Extracted Behavioral Test Generation Modeling**
  - Extraction
- **A case study**
  - A behavioral model of sigma-delta converter
  - A structural model of sigma-delta converter
- **Results**
- **Conclusions**

# Generic Test Generation Modeling

## Principle

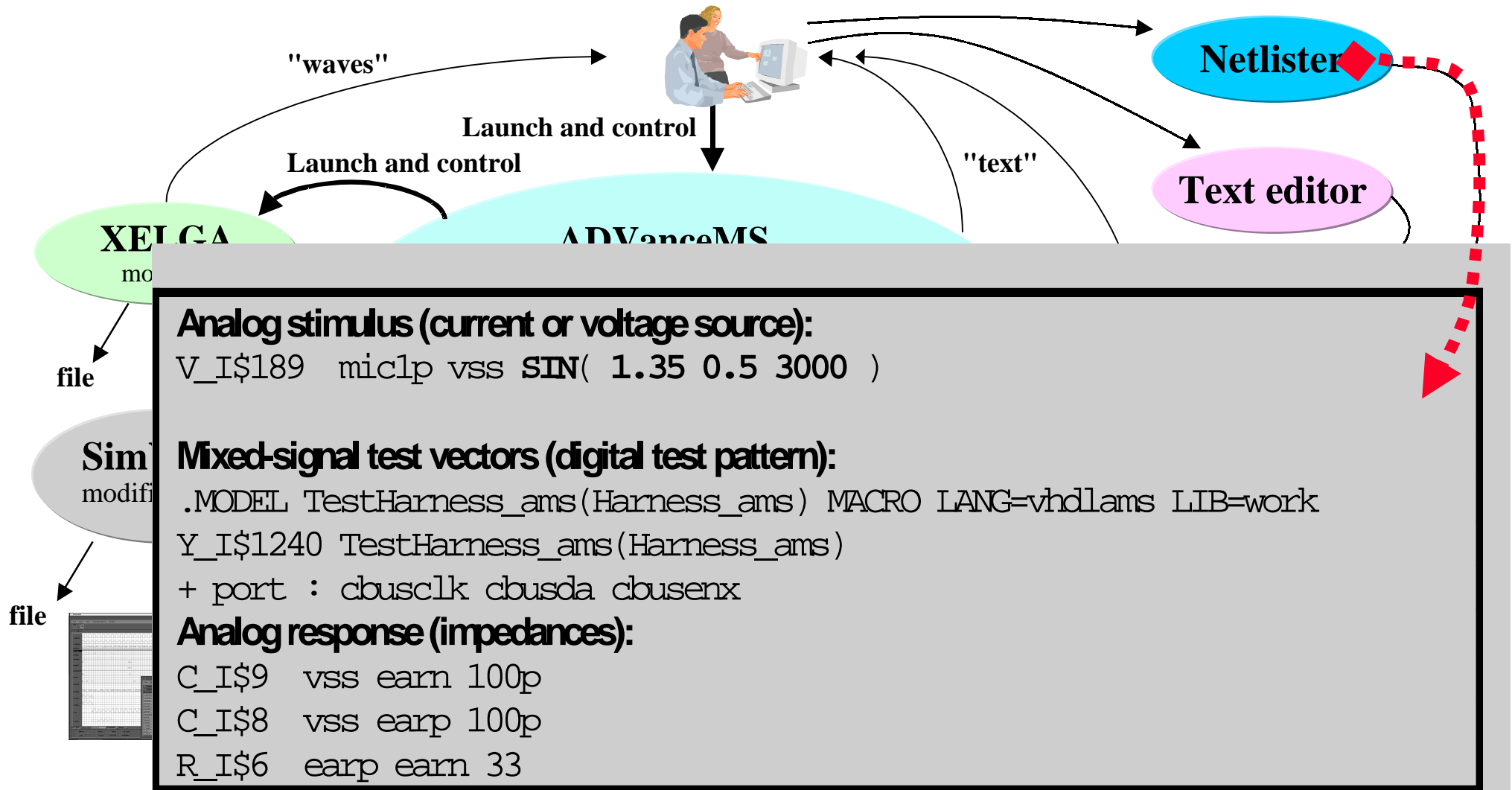


# Generic Test Generation Modeling Framework



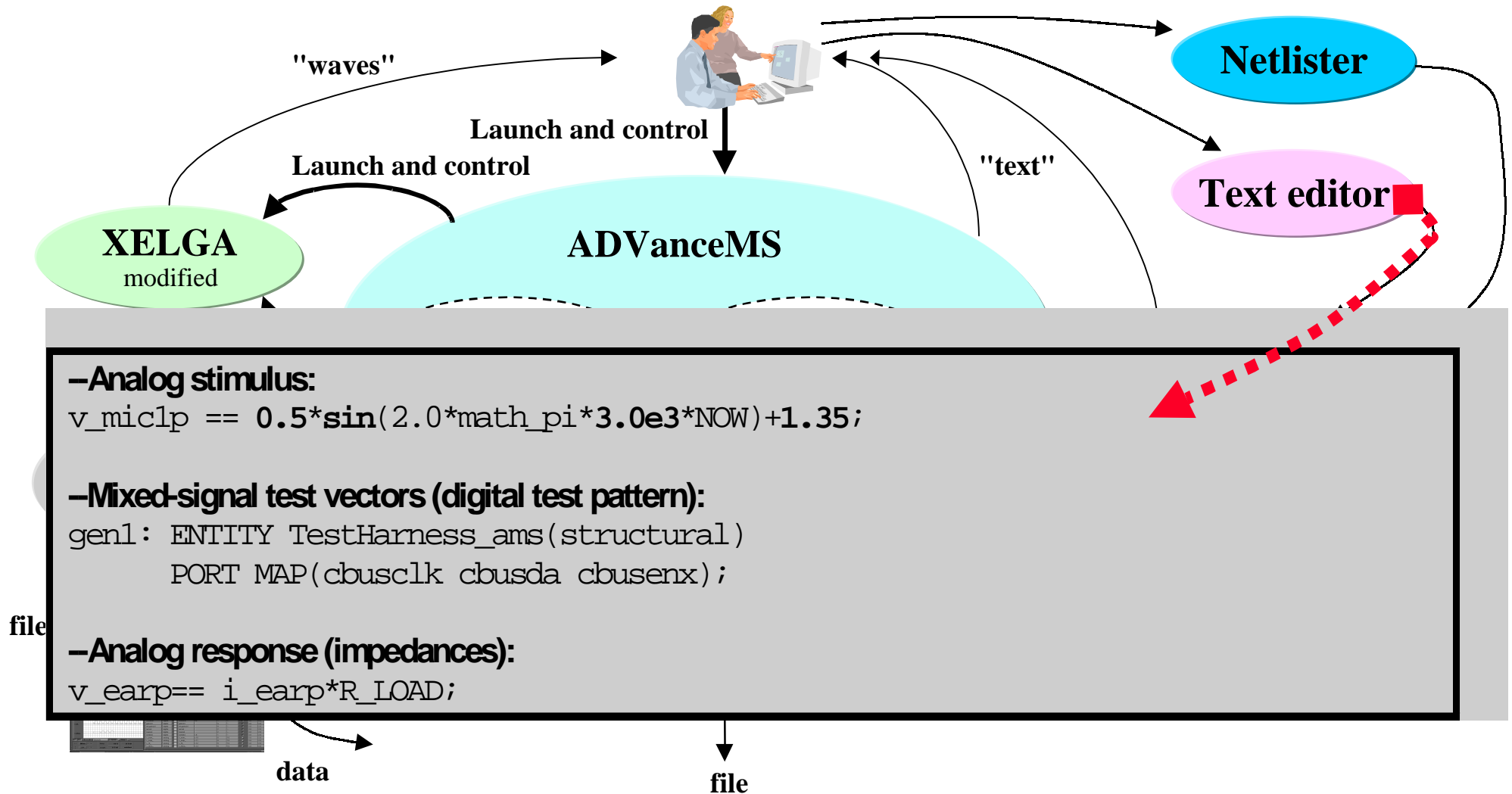
# Generic Test Generation Modeling

## Mechanism of SPICE netlist



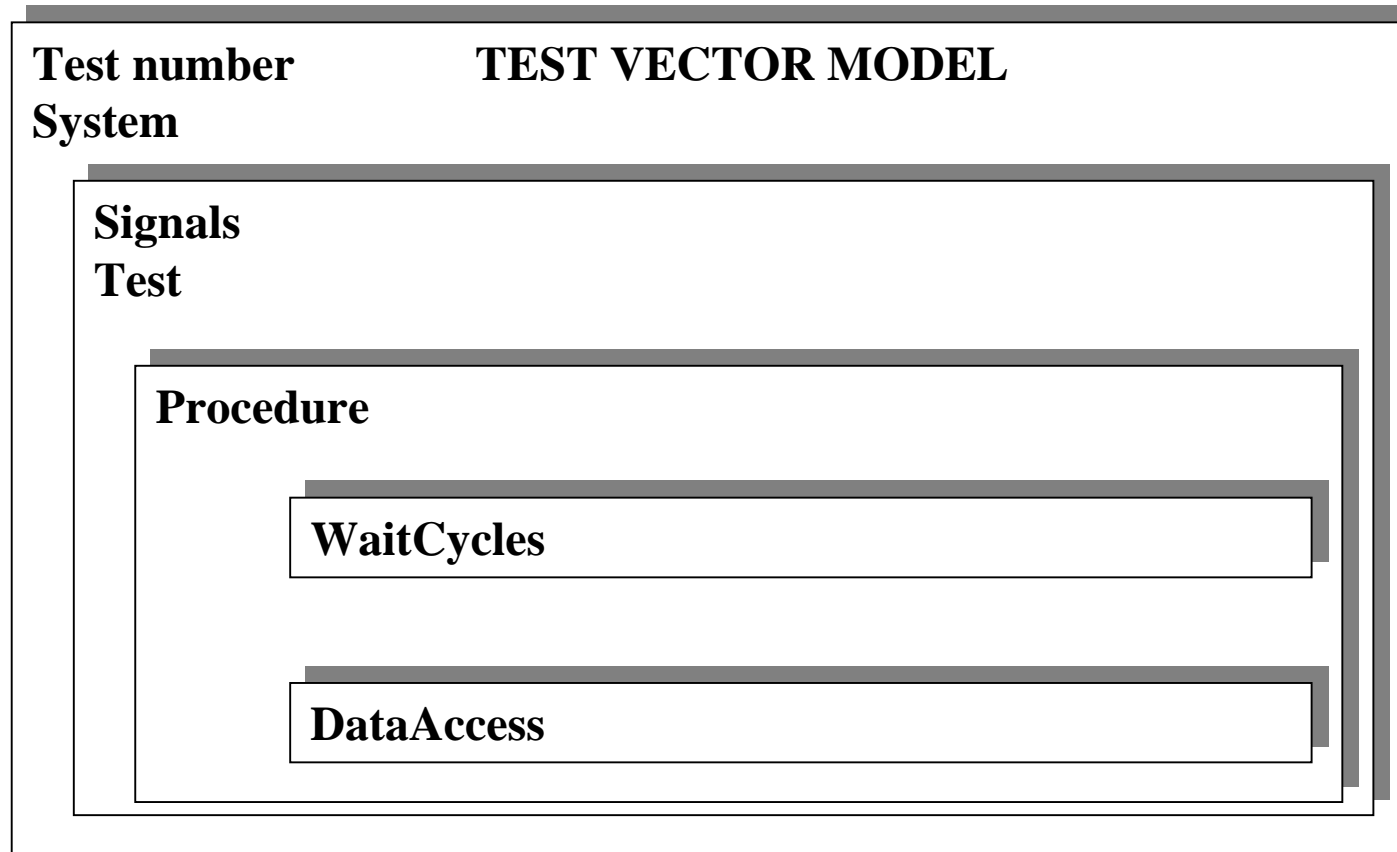
# Generic Test Generation Modeling

## Mechanism of VHDL-AMS description



# Generic Test Generation Modeling

## Test vector model



# Generic Test Generation Modeling

## Test vector model(cont'd)

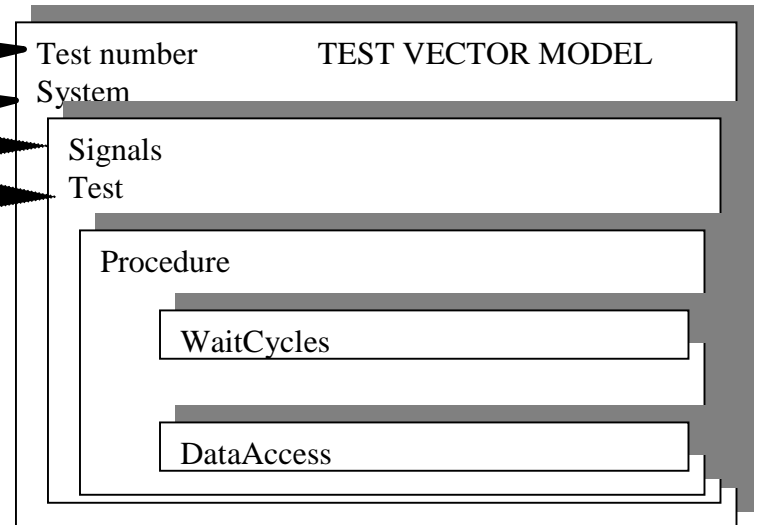
Entity (testbench entity)

```
ENTITY TestHarness IS  
  GENERIC (Test : NATURAL := 2;  
           System: T_System := GSM);
```

```
END ENTITY TestHarness;
```

Architecture (test definition, test signals)

```
SIGNAL AuxD STD_ULOGIC;  
...  
CASE Test IS  
  WHEN 1 => NULL;  
  WHEN 2 => TestAudioConverters(CBusClk, CBusDa, CBusEnX, Dump, CbusStatus);  
  WHEN OTHERS => NULL;  
END CASE;  
...
```



# Generic Test Generation Modeling

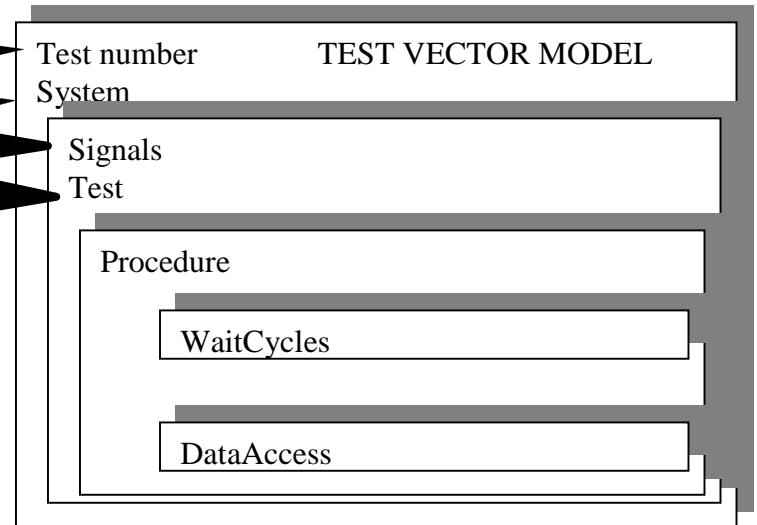
## Test vector model (cont'd)

Entity (testbench entity)

```
ENTITY TestHarness IS  
  GENERIC (Test : NATURAL := 2;  
          System: T_System := GSM);  
END ENTITY TestHarness;
```

Architecture (test definition, test signals)

```
SIGNAL AuxD STD_ULOGIC;  
...  
CASE Test IS  
  WHEN 1 => NULL;  
  WHEN 2 => TestAudioConverters(CBusClk, CBusDa, CBusEnX, Dump, CbusStatus);  
  WHEN OTHERS => NULL;  
END CASE;  
...
```



# Generic Test Generation Modeling

## Test vector model (cont'd)

Package (test procedure declaration)

```
PACKAGE TestPackage IS
...
PROCEDURE TestAudioConverters (SIGNAL Clock      : IN      STD_ULOGIC;
                               SIGNAL DataLine: INOUT  STD_LOGIC;
                               SIGNAL EnableX   : OUT   STD_ULOGIC;
                               SIGNAL DatumOut  : OUT   STD_ULOGIC_VECTOR;
                               SIGNAL Status    : OUT   T_Status);
...

```

Package body (test procedure: timing and register controls)

```
Package body TestPackage is
...
PROCEDURE TestAudioConverters (SIGNAL Clock      : IN STD_ULOGIC;
                               SIGNAL DataLine: INOUT STD_LOGIC;
                               SIGNAL EnableX   : OUT  STD_ULOGIC;
                               SIGNAL DatumOut  : OUT  STD_ULOGIC_VECTOR;
                               SIGNAL Status    : OUT  T_Status) IS

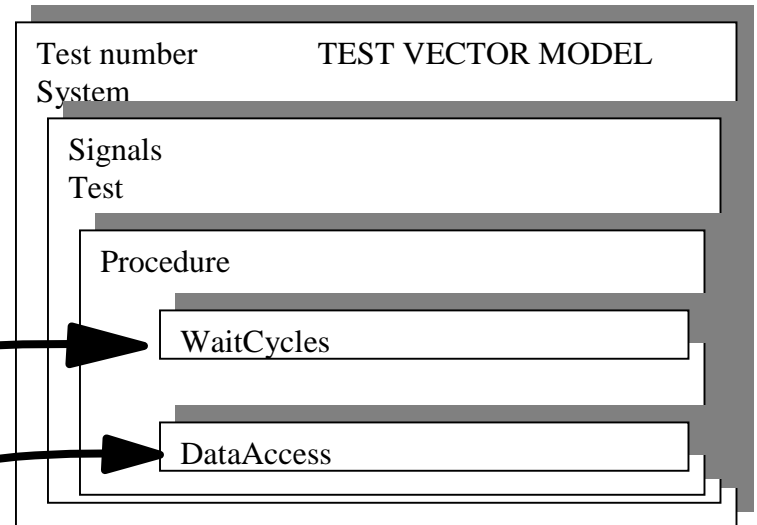
BEGIN

-- Identify the test to the simulator.
ASSERT (False)
  REPORT "Running audio converters test 2."
  SEVERITY NOTE;
WaitCycles (Status, Clock, 100, '0');
DataAccess (Chip, WriteDatum, AudioEnCtrl1R, "XX00000000000000",
            Clock, DataLine, EnableX, DatumOut, Status);

WaitCycles (Status, Clock, 10, '0');
DataAccess (Chip, WriteDatum, AudioEnCtrl1R, "XX10010000000000",
            Clock, DataLine, EnableX, DatumOut, Status);

WaitCycles (Status, Clock, 500, '0');
...

```



# Extracted behavioral test generation

## SPICE netlist testbench

SPICE netlist

```
V_I$189 mic1p vss SIN( 1.35 0.5 3000 )
```

```
C_I$9 vss earn 100p
```

```
C_I$8 vss earp 100p
```

```
R_I$6 earp earn 33
```

- Automatic generation

# Extracted behavioral test generation

## VHDL-AMS coded testbench

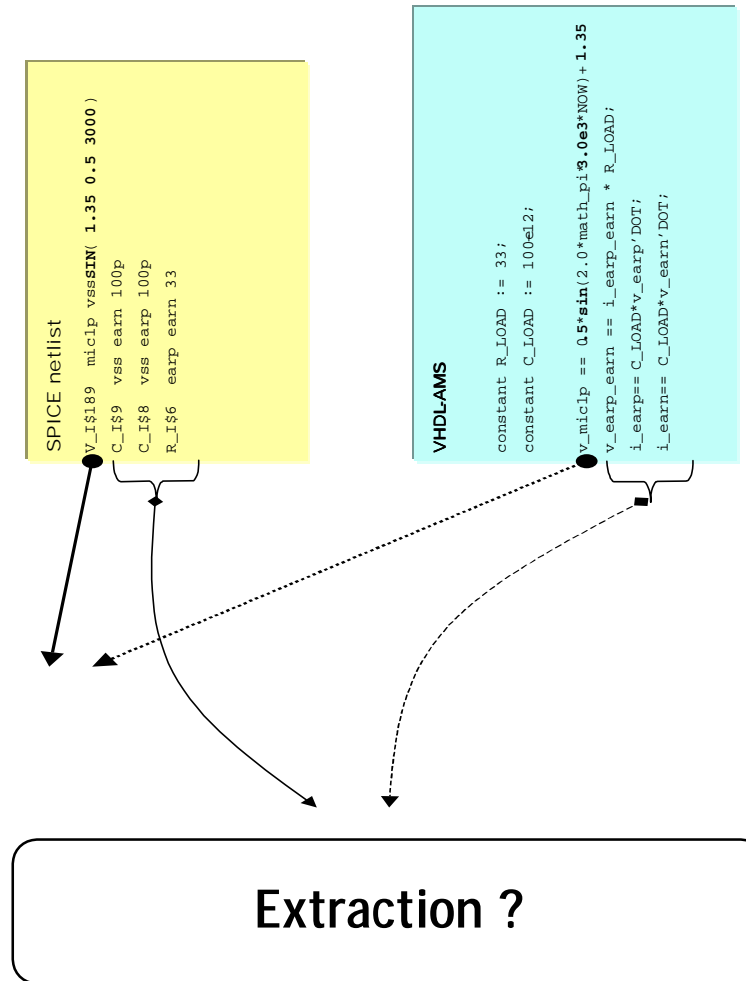
VHDL -AMS

```
constant R_LOAD := 33;  
constant C_LOAD := 100e - 12;  
  
v_miclp ==      0.5*sin (2.0*math_pi*      3.0e3 *NOW)+ 1.35 ;  
v_earp_earn == i_earp_earn * R_LOAD;  
i_earp==      C_LOAD*v_earp'DOT;  
i_earn==      C_LOAD*v_earn'DOT;
```

•Manual generation

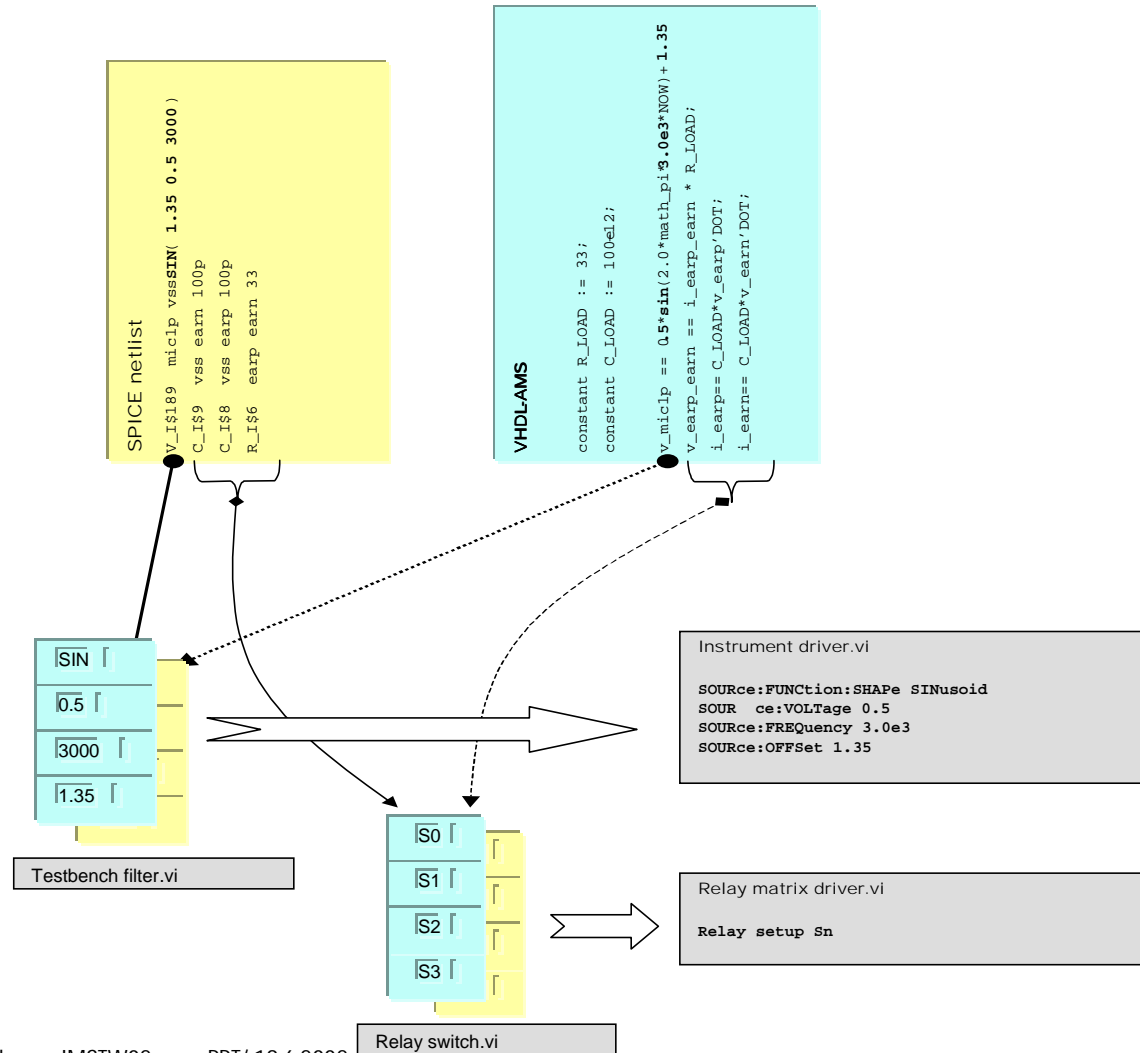
# Extracted behavioral test generation

## Extraction



# Extracted behavioral test generation

## Extraction



# A case study

## Behavioral model of sigma-delta converter

```

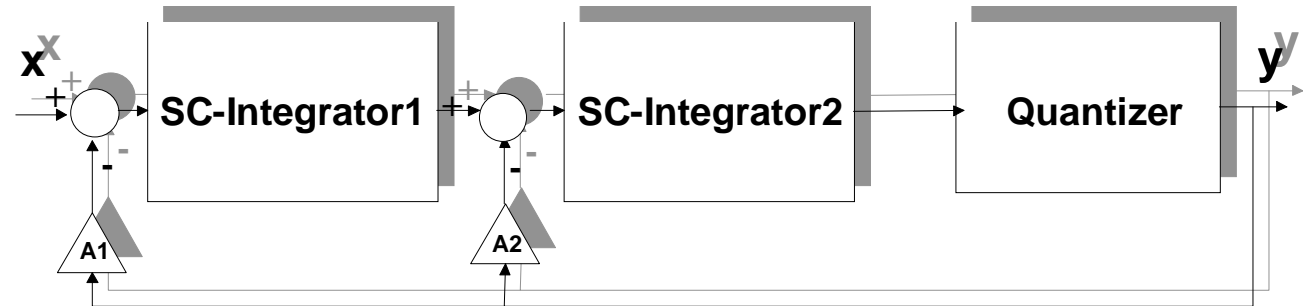
ENTITY sdm_mod IS
  GENERIC (A: REAL := 1.0;
           B: REAL := 2.0;
           C: REAL := 2.0);
  PORT (SDMP ELECTRICAL;
        SDMM ELECTRICAL;
        Y REAL := 0.0);
  ...

```

```

ARCHITECTURE behavior OF sdm_mod IS
  QUANTITY X across SDMP to SDMM;
  ...

```



```

  ...
  VARIABLE Stage1 : REAL := 0.0;
  VARIABLE Stage2 : REAL := 0.0;
  VARIABLE Comparator : REAL := -1.0;
  ...

  Stage1 := Stage1 + X - A * C * Comparator;
  Stage2 := Stage2 + Stage1 - B * C * Comparator;

  IF (Stage2 <= 0.0) THEN
    Comparator := -1.0;
    Y <= V_MINUS;
  ELSE
    Comparator := 1.0;
    Y <= V_PLUS;
  END IF;

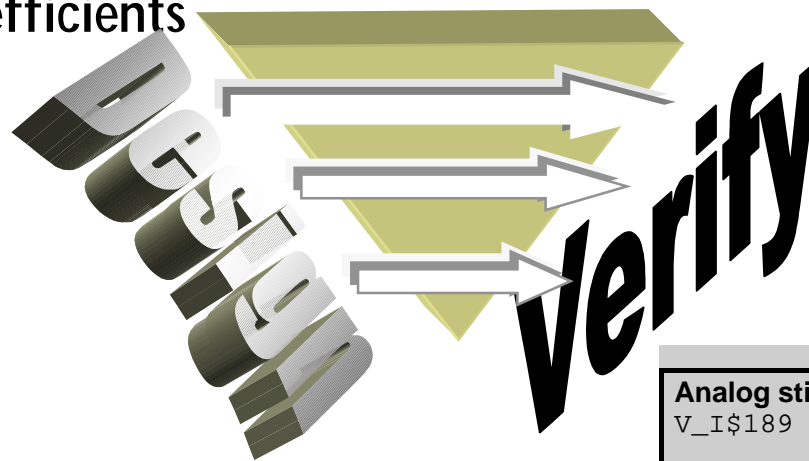
  ...
END ARCHITECTURE;

```

# A case study

## Behavioral model of sigma-delta converter (cont'd)

- tune feedback coefficients
- verify stability
- verify S/N figure



- *create test cases*
  - *specify test limits*
- *check test functionality*
- *create test vector patterns*
  - *run mixed-signal simulations*

### Analog stimulus (current or voltage source):

```
V_I$189 mic1p vss SIN( 1.35 0.5 3000 )
```

### Mixed-signal test vectors (digital test pattern):

```
.MODEL TestHarness_ams(Harness_ams) MACRO  
LANG=vhdlams LIB=work  
Y_I$1240 TestHarness_ams(Harness_ams)  
+ port : cbusclk cbusda cbusenx
```

### Analog response (impedances):

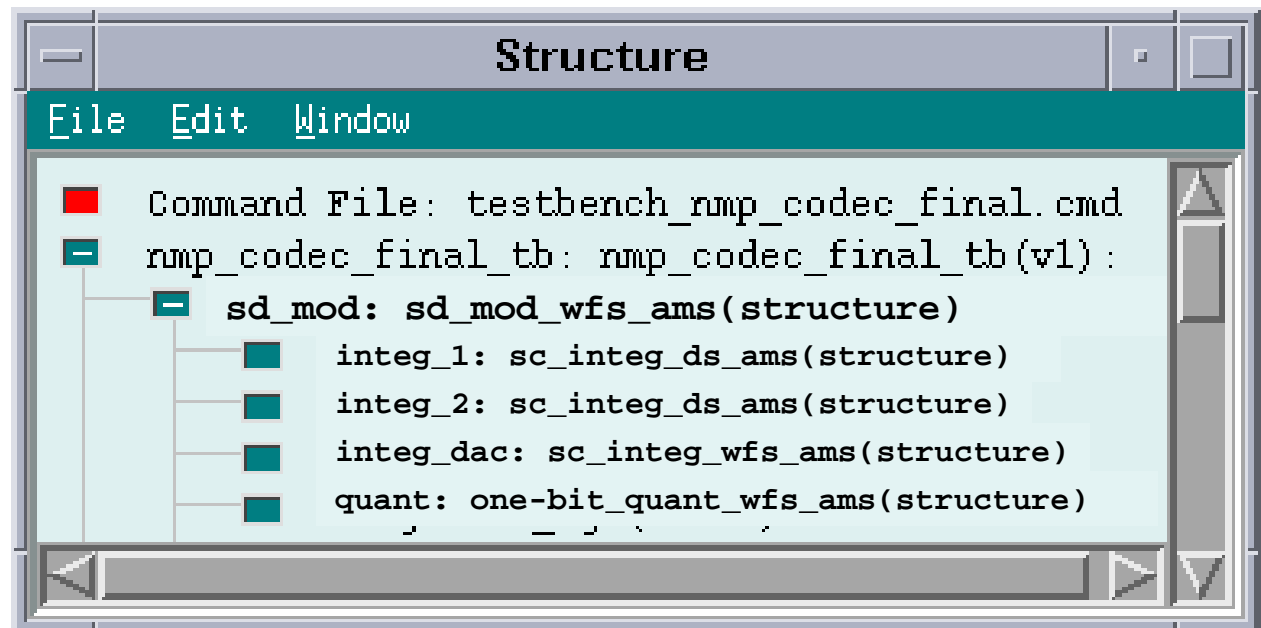
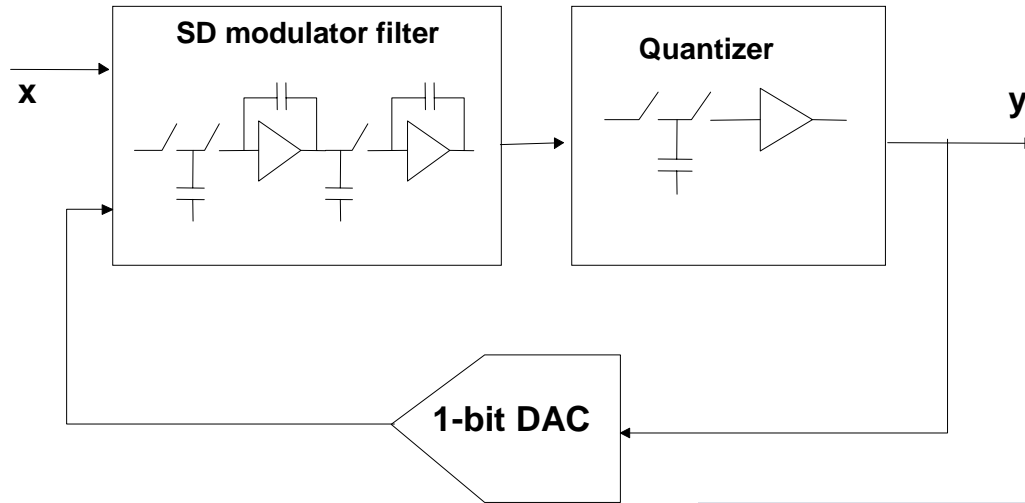
```
C_I$9 vss earn 100p  
C_I$8 vss earp 100p  
R_I$6 earp earn 33
```

### DUT:

```
.MODEL CODEC_NMP(BEHAVIOR) MACRRO LANG=vhdlams  
LIB=work  
Y_I$100 CODEC_NMP(BEHAVIOR)  
+port : vdaudli vsaudli mic1p mic1n micdata
```

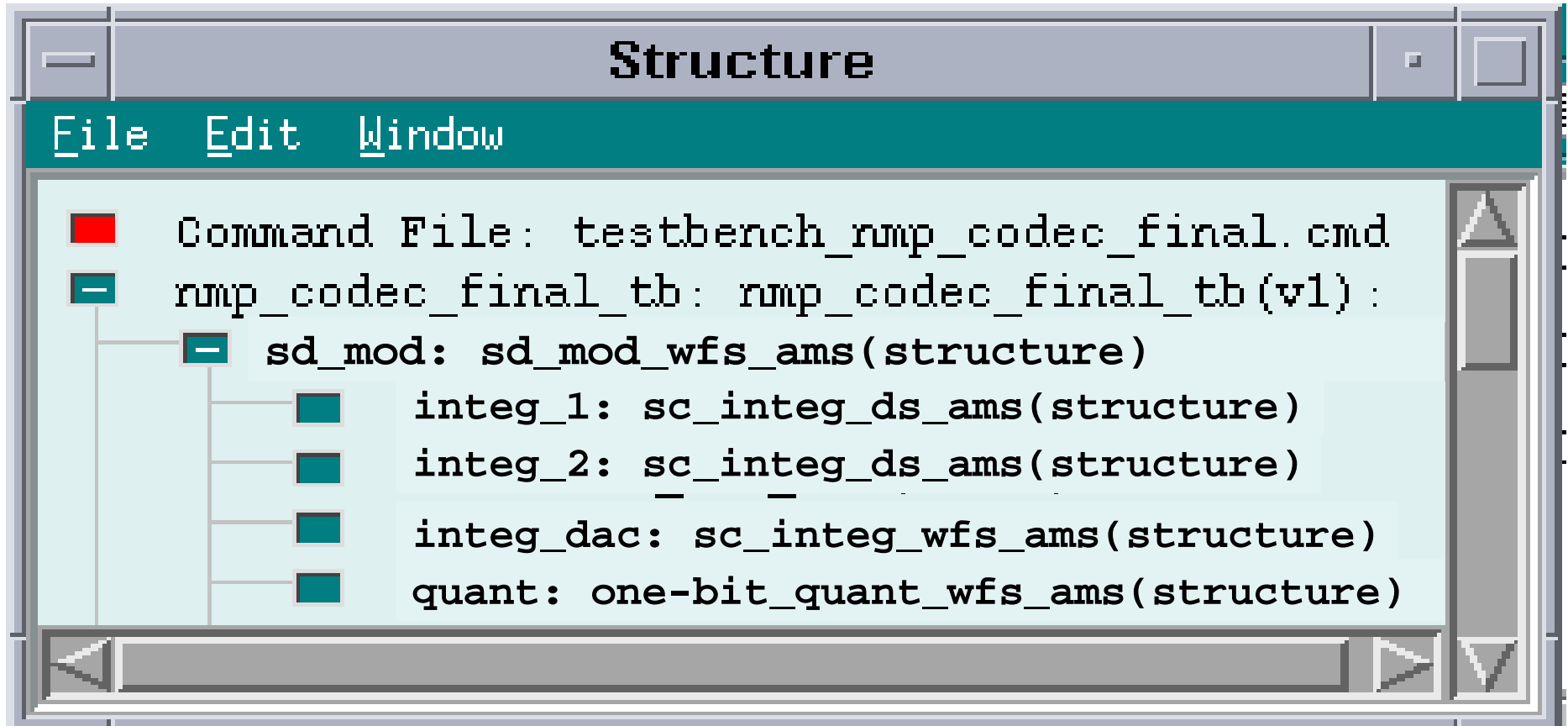
# A case study

## Structural model of sigma-delta converter (cont'd)

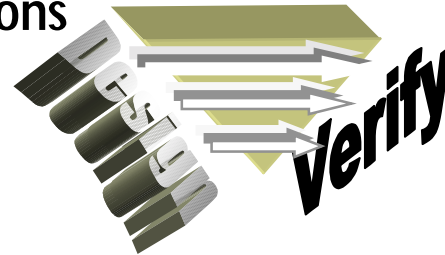


# A case study

## Structural model of sigma-delta converter (cont'd)



- find amplifier and quantizer specifications
- check clocking and timing
- find switch parameters



- *check test limits*
- *check test vector patterns*

# A case study

## Typical structural parameters

integ\_1  
integ\_2  
integ\_dac

- Sampling capacitance
- Integrator capacitance
- Switch on resistance
- Switch parasitic capacitance
- OTA parameters:

gm  
Rout  
Cin  
Cout  
I

quantizer

- Latch
  - propagation delay
- Comparator
  - OTA parameters

# Results

- Test specification = extracted test generation model(s) and test limits
- Testbench- top-level description of device:
  - Schematic based testbench - a SPICE netlist
  - A VHDL-AMS based testbench –a VHDL-AMS code
- Mixed-signal behavioral modelling is very suitable for test generation:
  - VHDL-AMS models are accurate
  - Available in early design phase
  - Simulation times are reasonable
  - Behavioral models are able to describe the silicon and real interface signals

## *Performance figures: simulation time and S/N ratio*

	Accusim	PowerMILL ACE	AdvanceMS Behavioral	AdvanceMS Structural
Block level	28 days – 79.52 dB	26.4 h – 51.5 dB	11 min – 86.5 dB	1.1 h – 81.5 dB
Top-level	NA	48 h	5.3 h	16.0 h

# Results (cont'd)

- Reuse of design and verification
  - different test platforms, different design environments
- Aid for test vector setup, test debug and verification

<b>Task</b>	<b>Design Engineer</b> Now / Savings [my=man-year]	<b>Test Engineer</b> Now / Savings [my=man-year]
Design Specifications	6 my / - 1 my	0 my / 0 my
Test Plan Design	0 my / +0.5 my	1 my / -0.5 my
Design & Simulate	8 my / 0 my	0 my / 0 my
Test System&Test Program Design	0 my / +0.5 my	2 my / -1 my
Test system verification	0 my / +0.2 my	1 my / 0 my
Prototype Testing	0 my / 0 my	1-2 my / 0 my
<b><i>Design development gap</i></b>	<b><i>1 my / -1 my</i></b>	
<b><i>Test development gap</i></b>		<b><i>1 my / -1 my</i></b>
<b><i>ALL TOGETHER</i></b>	<b><i>15 my / -0.8 my</i></b>	<b><i>5-6 my / -1,5 my</i></b>

# Results (cont'd)

- Reuse of design and verification
  - different test platforms, different test environments
- Aid for test vector setup, test debug and verification

**Total savings 13 %**  
**2.3 manyears**  
**• 18 manyear project**

Task	Engineer	Savings [my=man-year]
Design Specifications		
Test Plan Design		
Design & Simulate		
Test System&Test Program		
Test system verification		1 my / 0 my
Prototype Testing		1-2 my / 0 my
<b>Design development gap</b>		
<b>Test development gap</b>		1 my / -1 my
<b>ALL TOGETHER</b>		5-6 my / -1,5 my

# Conclusions

- Generic behavioral test generation modeling is suitable for generating and handling the complexity of mixed-signal devices
- Configuring mixed-signal instruments into different testcases can be made in considerable time, new testcases are feasible to take account
- A change in design-to-test flow
- Total development time of mixed-signal chips is shortened

**Thank You for attention**