

HSPICE data structure refactoring: Rewrite Parser codes for some commands with C++

Intern: Hu, Hanbin

From Shanghai Jiao Tong University

Major In Microelectronics

Mentor: Yang, Guoyu & Zhu, Liping

Agenda

- Objective
 - Command Introduction
- Design Methodology
 - File Organization
 - Interface
- Results and Achievement
- TF engine Rewrite (Extra work)

Comparison between Fortran and C++

Fortran

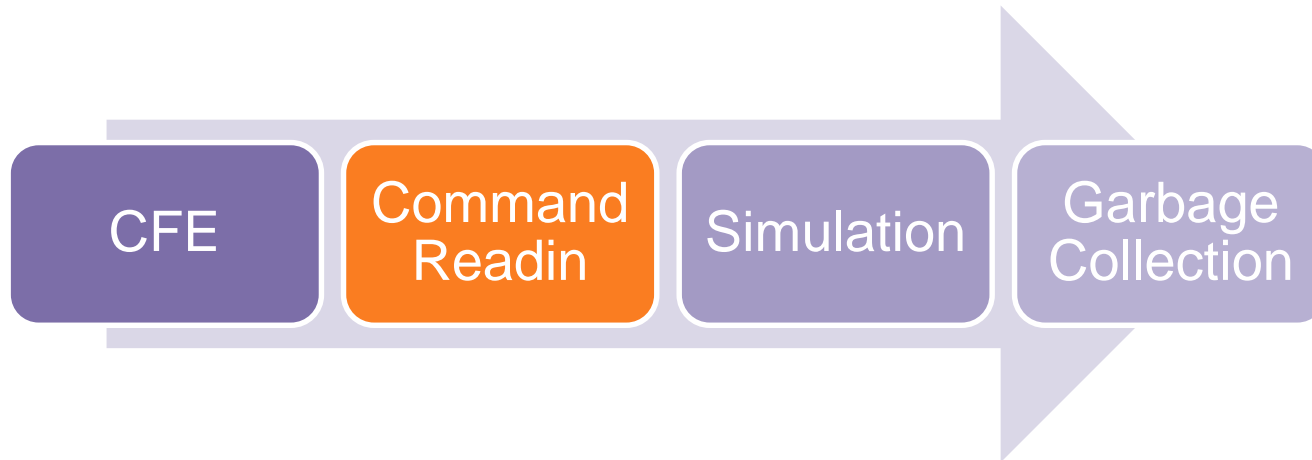
- Procedural-Oriented
- Data exposure
- Hard maintenance
- Low reusability
- goto command
- Easy to follow
- Quick Execution

C++

- Objected-Oriented
- Scope and encapsulation
- Namespace and class
- Object responsibility
- Well-designed interface
- Messed-up classes
- Slower due to overhead

Objective

- Translate Fortran code to C++ in command readin stage



- CFE provides formalized input and prevents obvious errors
- 'Command Readin' stage **properly stores** relevant information from CFE and **efficiently transfers** data to simulation engine.
- **Careful** garbage collection

Command Introduction

.set_sample_time command

- Syntax

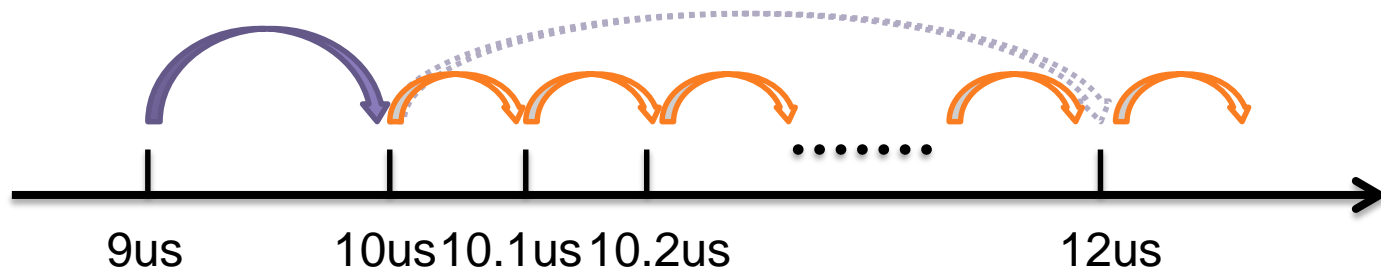
`.set_sample_time` **twindow** (*start_time stop_time* [*start_time stop_time*])
period *period_value*

- Function

- Forces HSPICE to compute the data points with **a fixed time step**.

- Example

`.SET_SAMPLE_TIME twindow 10u 20u period 100n`



Command Introduction

.dout command

- Syntax

`.dout node vth (time state [time state])`

`.dout node vlo vhi (time state [time state])`

state	Meaning
0	Low
1	High
Others	Don't care

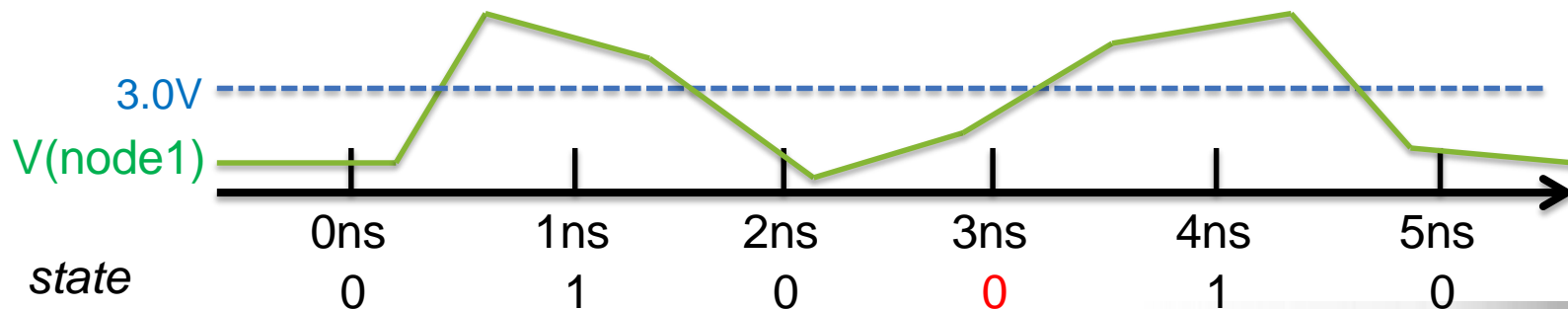
- Function

– Specifies the **expected** final state of an output signal.

- Example

`.param VTH=3.0`

`.dout node1 VTH (0.0n 0 1.0n 1 2.0n X 3.0n 1 4.0n Z 5.0n 0)`



Command Introduction

.store command

- Syntax

```
.store [type=ic|nodeset|memdump] [file=save_file_prefix] [trantime=0/1]  
[time=time1][time=time2]...[time=timeN] [repeat=period] [save_on_kill=0/1]
```

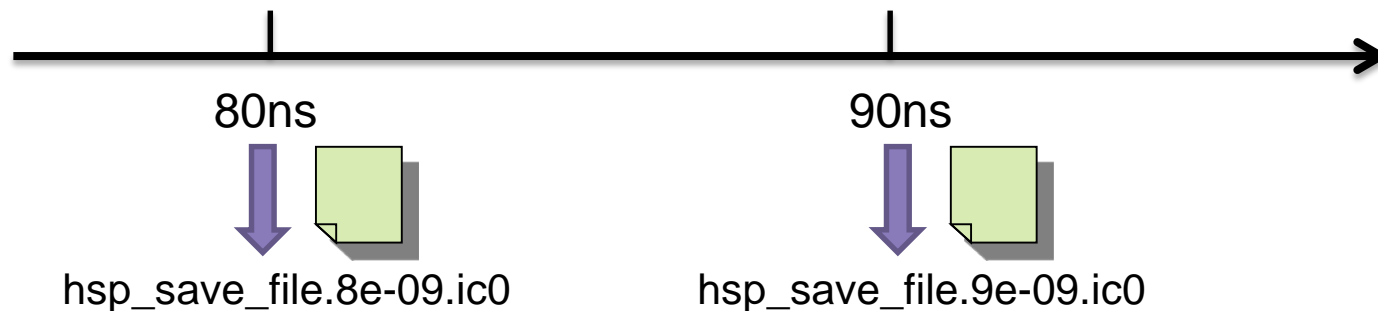
- Function

- Starts a store operation to **create checkpoint files** describing a running process during transient analysis.

- Example

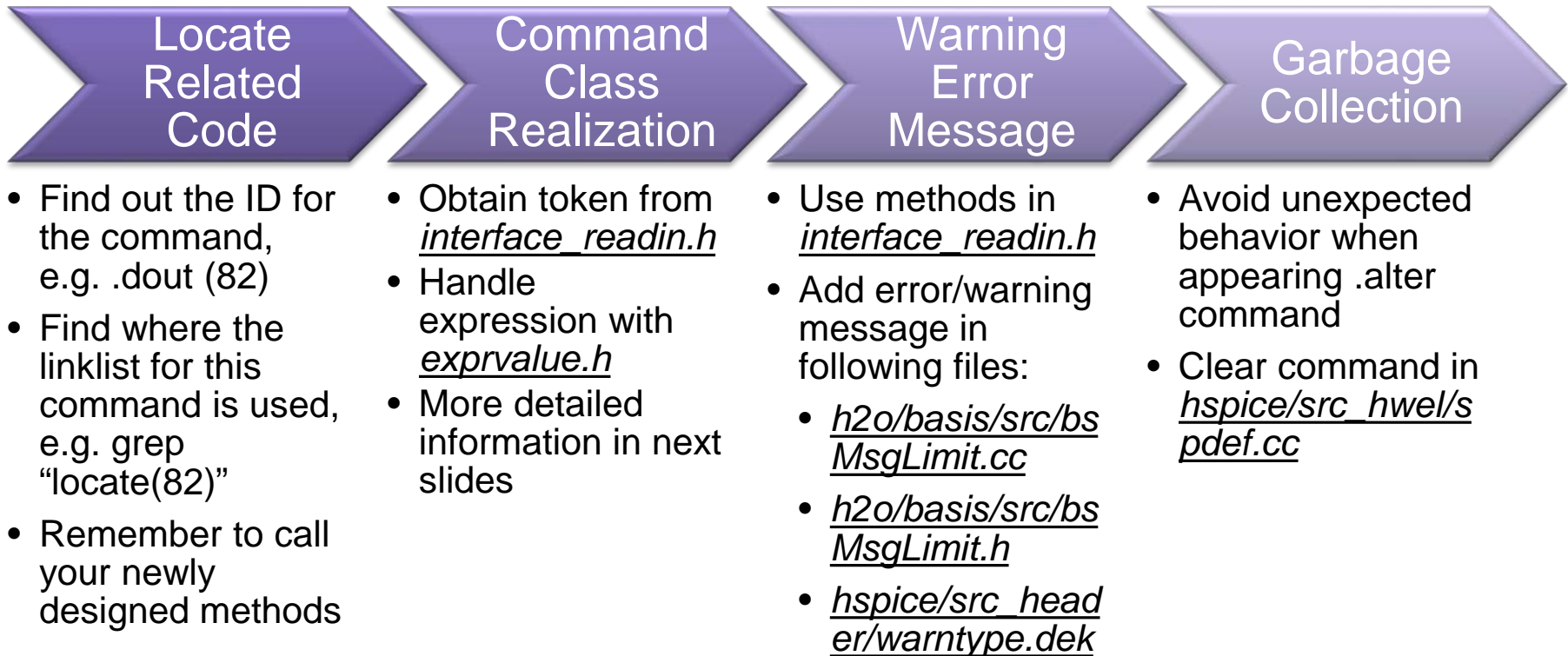
Only last statement takes effect.

```
.store time=80n time=90n file="hsp_save_file"
```



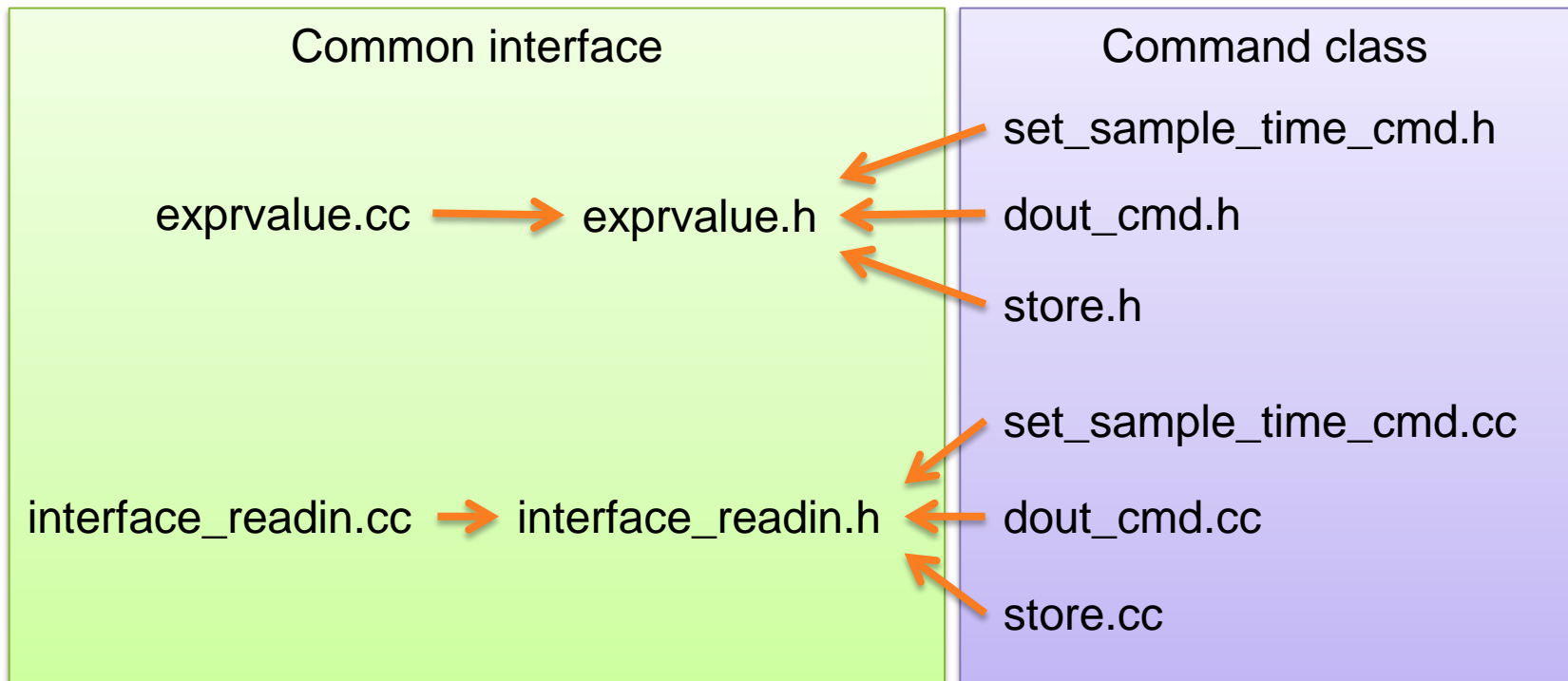
Work flow to translate new command

- Overall procedure



File Organization

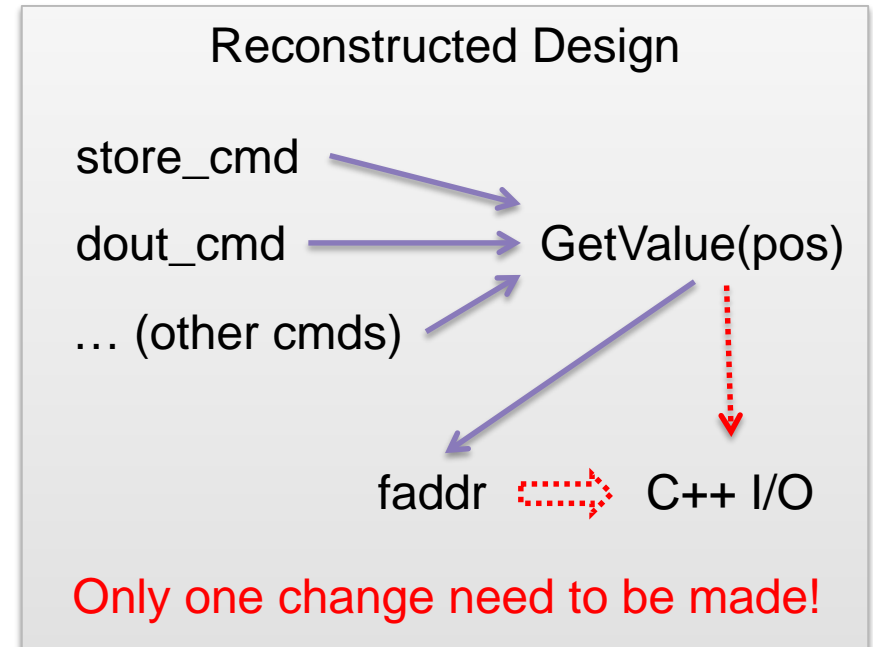
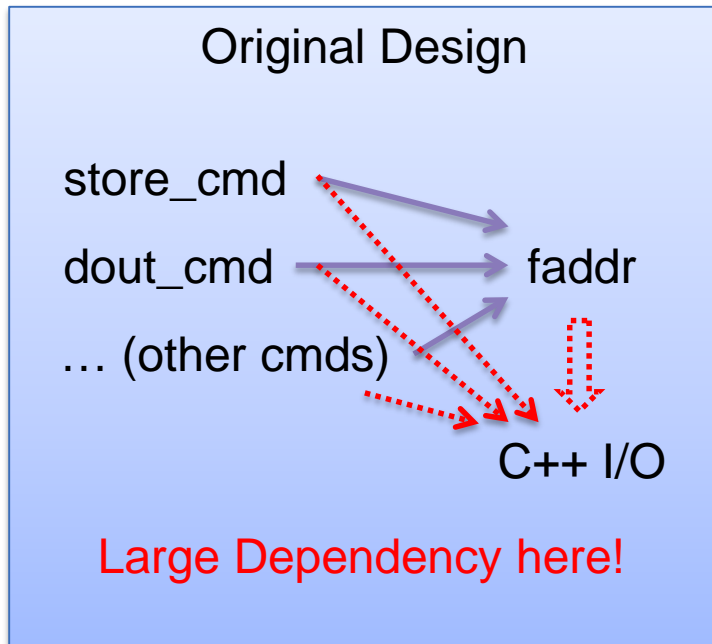
- All translation files newly added are stored in folder 'hspice/src_commands.'



Common Interface

Concept (Cont'd)

- Reduce dependency
- Newly designed structure



Common Interface

Implementation

- Error / Warning message
 - **Unified** output format
 - Error and warning interface for **readin and usage phase**
- Obtain token from linklist in Fortran
 - code, value, name, delimiter
- Deal with Expression
 - Only consider **top-level** parameter for now, i.e., no subckt.
 - Get / Set method

```
double GetValue();
```

```
void SetValue(double value);
```

```
void SetValue(char* expr, double val, int code);
```

Peculiar interfaces between Fortran and C++ for each command are not included to avoid a **fat interface file**.

Source Code ChangeLog for Parser

14 files added:

- hspice/src_commands/Master.make
- hspice/src_commands/exprvalue.cc
- hspice/src_commands/exprvalue.h
- hspice/src_commands/interface_readin.cc
- hspice/src_commands/interface_readin.h
- hspice/src_commands/set_sample_time_cmd.cc
- hspice/src_commands/set_sample_time_cmd.h
- hspice/src_commands/dout_cmd.cc
- hspice/src_commands/dout_cmd.h
- hspice/src_commands/store_cmd.cc
- hspice/src_commands/store_cmd.h
- hspice/src_commands/ftn_instvalid.cc
- hspice/src_commands/instant_valid.cc
- hspice/src_commands/instant_valid.h

6 files removed:

- fortran/src_variation/sample_point.F
- fortran/src_cmd/readin_save_state.F
- fortran/src_cmd/readin_store.F
- hspice/src_hwel/ftn_instvalid.cc
- hspice/src_hwel/instant_valid.cc
- hspice/src_hwel/instant_valid.h

16 files modified:

- fortran/src_cmd/runcon.F
- fortran/src_base/readin_cmd.F
- fortran/src_base/errchk.F
- fortran/src_expr/parupd.F
- fortran/src_analysis/brkpbldr.F
- fortran/src_analysis/transtor.F
- fortran/src_analysis/hope_run_tr.F
- fortran/src_cmd/hope_readin_dout.F
- fortran/src_cmd/hope_output_cmd.F
- hspice/Master.make
- hspice/src_hwel/spdef.cc
- hspice/src_header/warntype.dek
- h2o/basis/src/bsMsgLimit.cc
- h2o/basis/src/bsMsgLimit.h
- io/output_adaptor/wave_signals.cc
- io/output_adaptor/ftn_message.cc

Run a regression with **8993 cases**: All pass!

TF Engine Introduction

- Syntax

`.tf ov srcname` ← DC: **transfer function**, **input** and **output** resistance

`.tf ac outval` ← AC: only transfer function

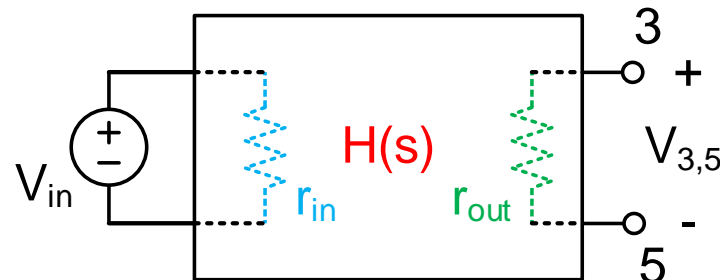
`.print ac tfv(Voltage_source_name) tfi(node_name)`

- Function

- Calculates small-signal values for transfer functions for both DC and AC simulations.

- Example

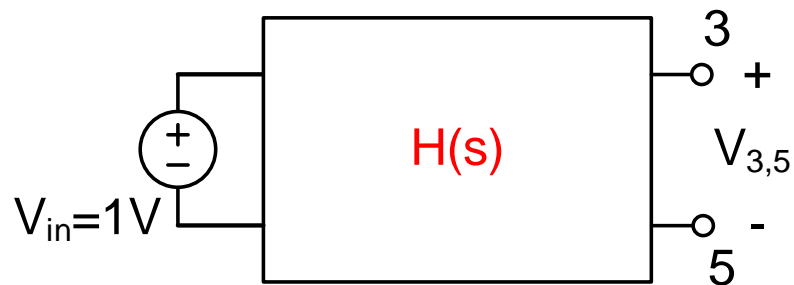
`.TF V(3, 5) VIN`



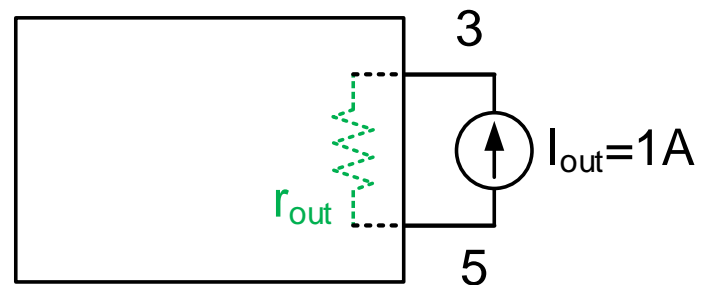
$H(s)$, r_{in} and r_{out} will be computed here.

TF Simulation Principle

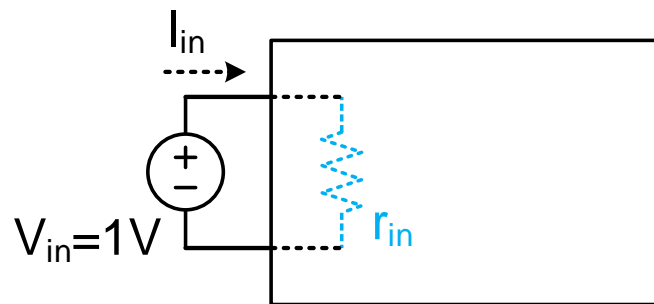
- Apply unity source according to input and output type



$$H(s) = \frac{V_{3,5}}{V_{in}} = V_{3,5}$$



$$r_{out} = \frac{V'_{3,5}}{I_{out}} = V'_{3,5}$$



$$r_{in} = \frac{V_{in}}{I_{in}} = \frac{1}{I_{in}}$$

Two matrix solve procedure for **three** required values

Source Code ChangeLog for TF

2 files added for tf dc:

hspice/src_op/hsp_dc_sstf.cc
hspice/src_op/hsp_dc_sstf.hh

2 files added for tf ac:

hspice/src_ac/hsp_ac_sstf.cc
hspice/src_ac/hsp_ac_sstf.hh

5 files modified for tf dc:

fortran/src_analysis/dc.F
fortran/src_analysis/dcsweep.F
fortran/src_base/num16.F
hspice/src_op/op_f2c.hh
hspice/src_op/op_linklist.cc

4 files modified for tf ac:

hspice/src_ac/ac_linklist.cc
hspice/src_ac/ac_linklist.hh
hspice/src_ac/ac_macro.hh
hspice/src_ac/acSmallSignal.cc

Run a regression with **8993 cases**: All pass!

Fixed bugs

.dout command output

- Wrong simulation output for multiple .dout commands (fixed)
 - Reasoning: **wrong node name stored**
 - Example case: DE_CASES/output/DE13098/DE13098_1.sp

```
.dout ss vlo vhi(6ms 1 16ms 0 ...)
```

```
.dout xcc.ds vth(1ms 0 1.5ms 0 ...)
```

Original HSPICE .lis file

```
*****Output vector error report*****  
output signal at node ss  
verified with no error.  
*****Output vector error report*****
```

Modified HSPICE .lis file

```
*****Output vector error report*****  
output signal at node ss  
verified with no error.  
*****Output vector error report*****  
+ output signal at node xcc.ds  
+ verified with no error.
```

- Unrealized multiple node feature (Manual correction)
 - Example

```
.dout B C D (0n 1 1 0 5n 0 0 0)
```


Fixed bugs

Miscellany

- Error / Warning output format disunity (fixed)

– Example

```
**warning** (save_period_01.sp:10) For multiple .store statement,the last one take effect.
```

warning
/ error

Simulation
file name

Line
number

Warning
message

- .store warning for last statement when .alter command appears (fixed)
- Missing list value in .instvalid command (fixed)
- Alignment in .tf dc output (fixed)

Conclusion

- Complete the translation from Fortran to C++ for the three commands (.set_sample_time / .dout / .store)
- Provide **common interface** for easy translation
- Design the software **architect and construction** for command readin
- Propose a **translation methodology** from Fortran to C++ for other commands
- Rewrite engine for transfer function simulation
- Detect and fix 6 bugs in HSPICE

Self Assessment

- **Second** year of internship in Synopsys
- **Collaborated** with talented and warm-hearted colleagues
- Contributed codes to a **mature commercial product** HSPICE
- Developed **large-scale** software system **first time**
- Learnt a programming language **FORTRAN**



Thank You

SYNOPSYS[®]
Accelerating Innovation

