A scalable SystemC stimuli parser for Digital and AMS simulations in COSIDE

Alessandro Putortì <alessandro.putorti@infineon.com>

4th Dec 2024

Table of contents

Table of contents

Stimulus Parser

```
STIM

# Ramp "VCC1" to 3.3V in 2us #
RAMP_VCC1 3.3 2

# Immediately set VCC2 to 3.0V
SET_VCC2 3.0

# WAIT for ramp on VCC1 to end
WAIT_RAMPS

# Set "SUPPLY_OK"
SET_SUPPLY_OK 1

# WAIT 10 ns #
WAIT_NS 10

# Set "DIN" to 0xA2 #
SET_DIN 0xA2

# Ramp "VCC2" to 0V in 2us #
RAMP_VCC2 0 2

# Check whether "check_valid" is '1' #
CHECK_VALID 1

# WAIT 10 ns #
WAIT_NS 10

# Read register 0x03 through i2c IF #
# and check whether it matches 0x17 #
I2C_READ 0x03 0x17

EXIT
```

stimulus.txt

Stimulus file

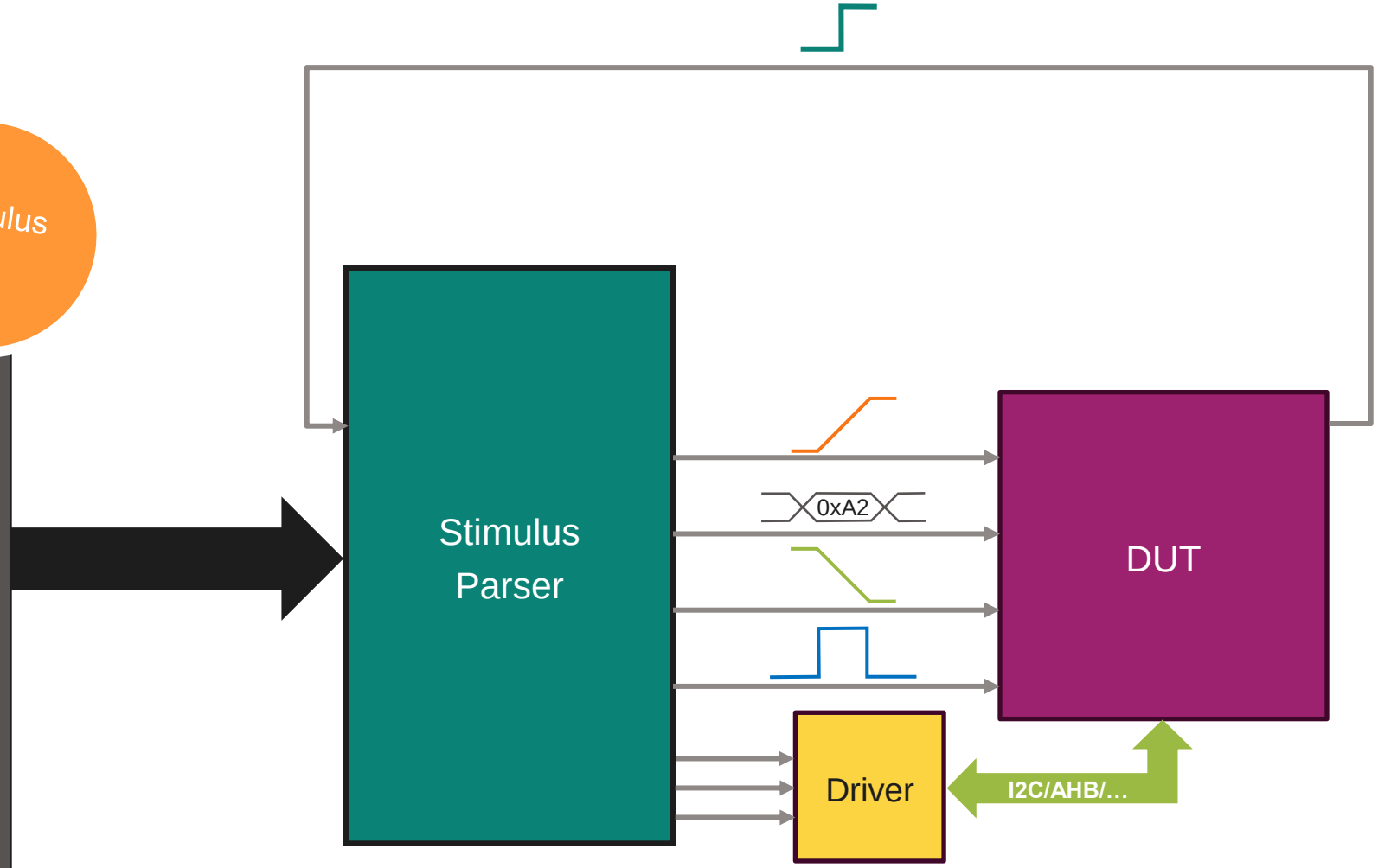
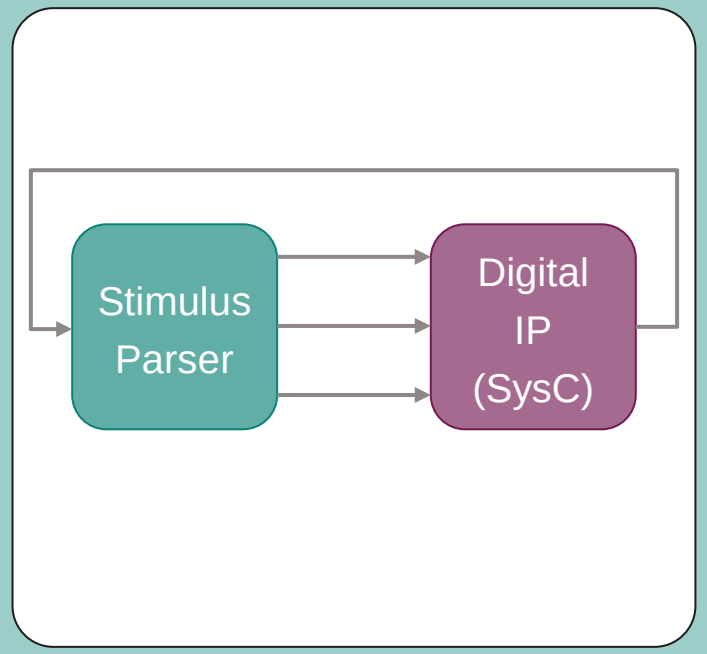


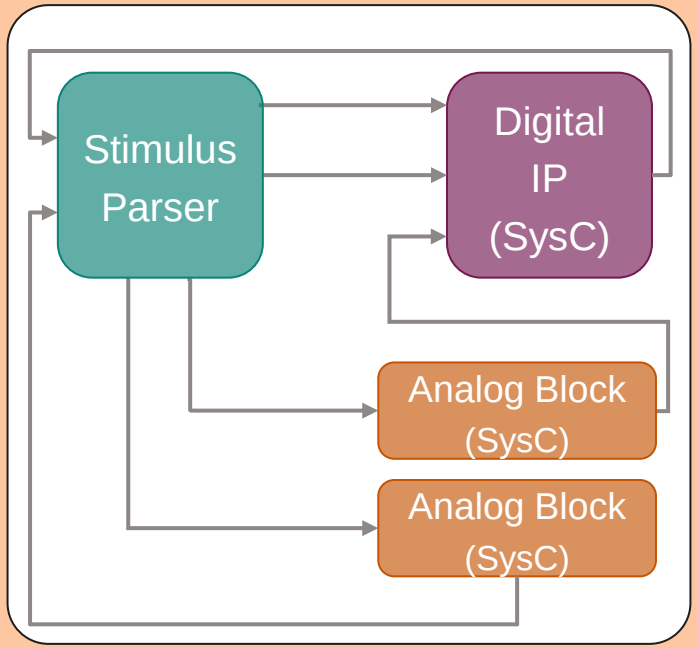
Table of contents

Common Scenarios



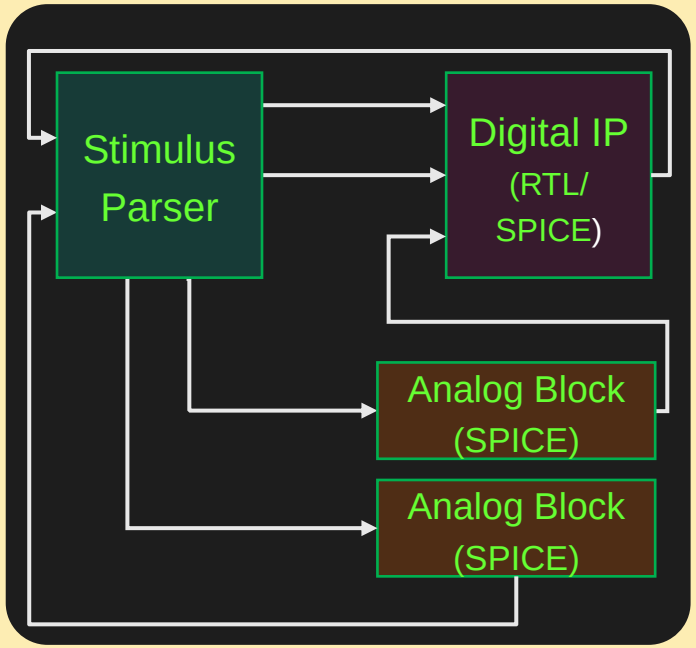
Scenario 1: SystemC Digital simulation

- Only digital ports (Boolean or buses)
- **Coside** environment



Scenario 2: SystemC AMS simulation

- Both digital and analog (**double**) ports
- **Coside** environment

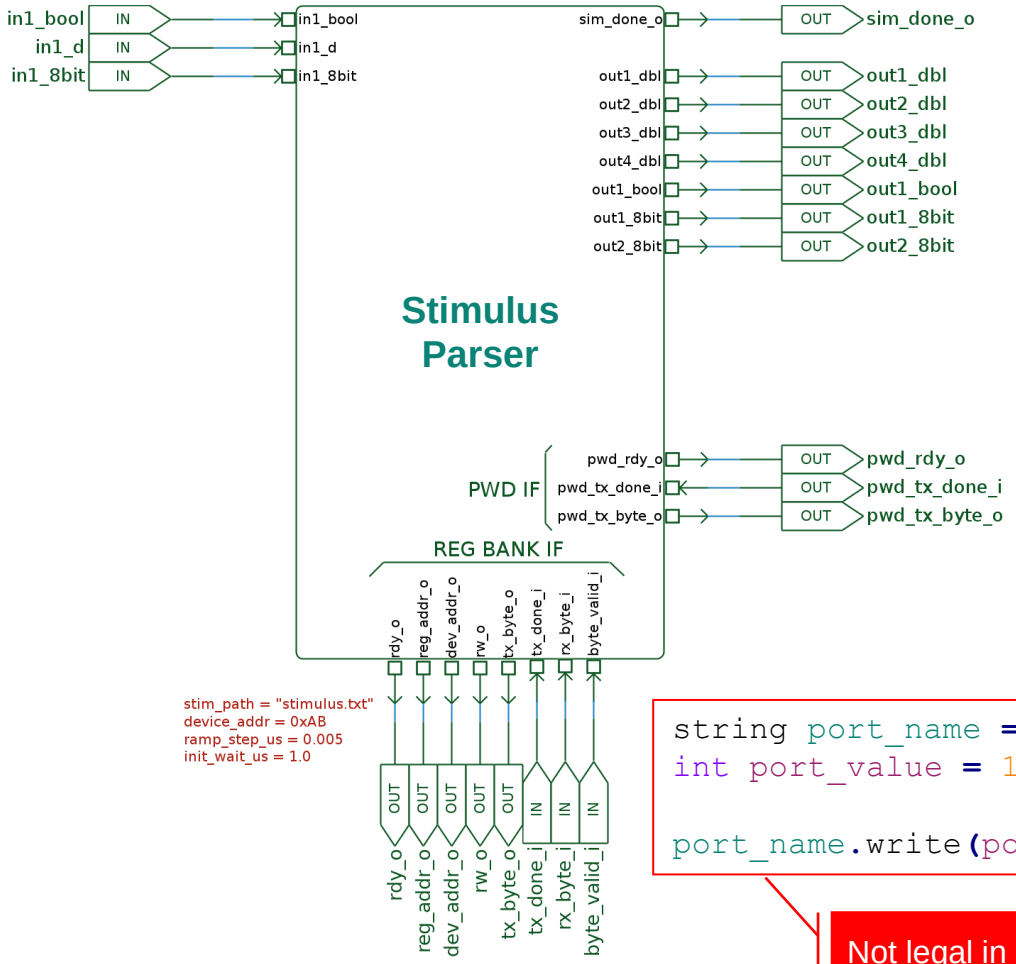


Scenario 3: Spectre AMS simulation

- Both digital and analog (**wreal**) ports
- **Cadence** virtuoso environment
- Multi-language scenario

Table of contents

Legacy Solution




```
string port_name = "out1_bool";
int port_value = 1;

port_name.write(port_value);
```

Not legal in C++ / SysC

```
int stim_parser_legacy::parse_line(std::string com,
std::string args){
    int ret=0;
    if(com.compare("SET_SIM_DONE_O")==0){
                    //convert args, check range, set
port "sim_done_o"
        ret=set_sim_done_o(args);
    }else if(com.compare("SET_OUT1_DBL")==0){
        ret=set_out1_dbl(args);
    }else if(com.compare("RAMP_OUT1_DBL")==0){
        ret=ramp_out1_dbl(args);
    }else if(com.compare("SET_OUT2_DBL")==0){
        ret=set_out2_dbl(args);
    }else if(com.compare("RAMP_OUT2_DBL")==0){
        ret=ramp_out2_dbl(args);
          ...

    }else if(com.compare("SET_OUT1_8BIT")==0){
        ret=set_out1_8bit(args);
    ...
    }else{
        ret=-2;
    }
    return ret;
}
```

You need to manually adapt the C++ code whenever you want to change the interface (number, name, type of ports):

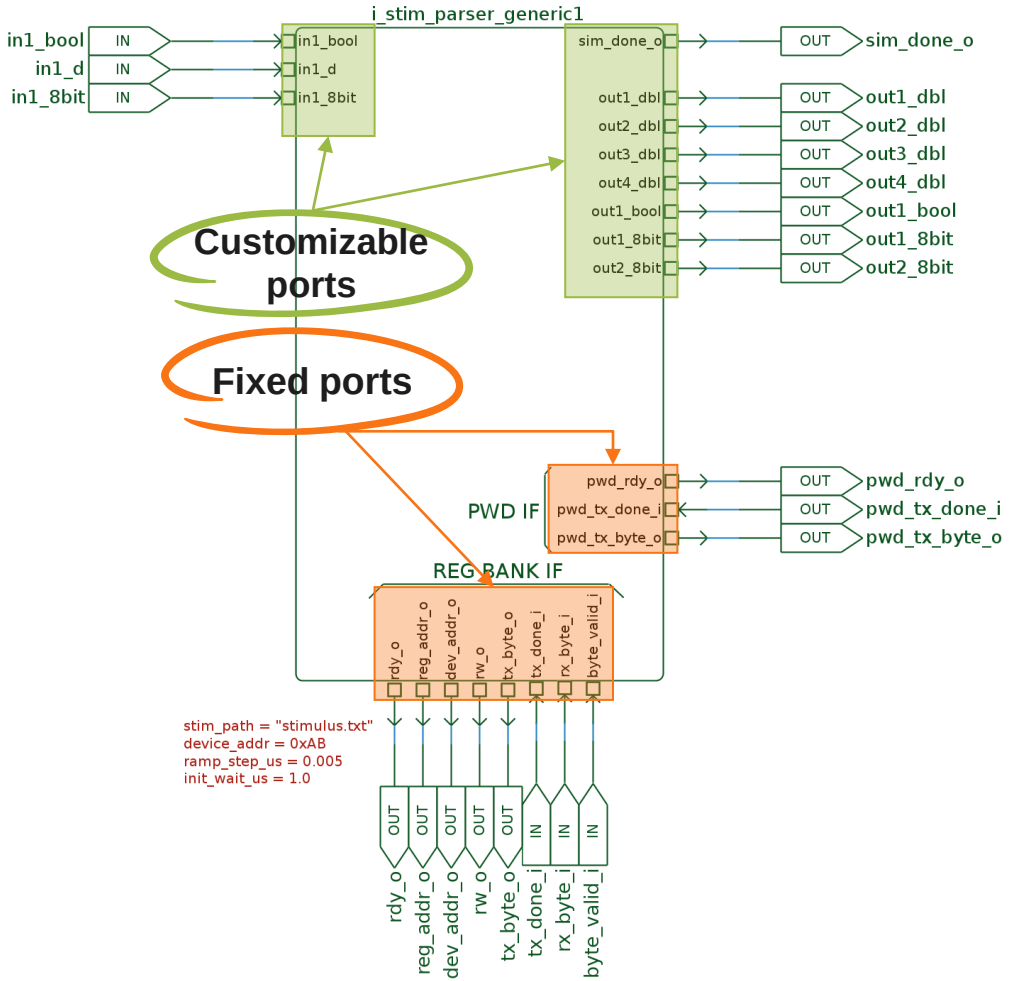
- **Time consuming**
- **Error prone**

Challenge: Make it reusable

Table of contents

Table of contents

Generic Stimulus Parser - Interface

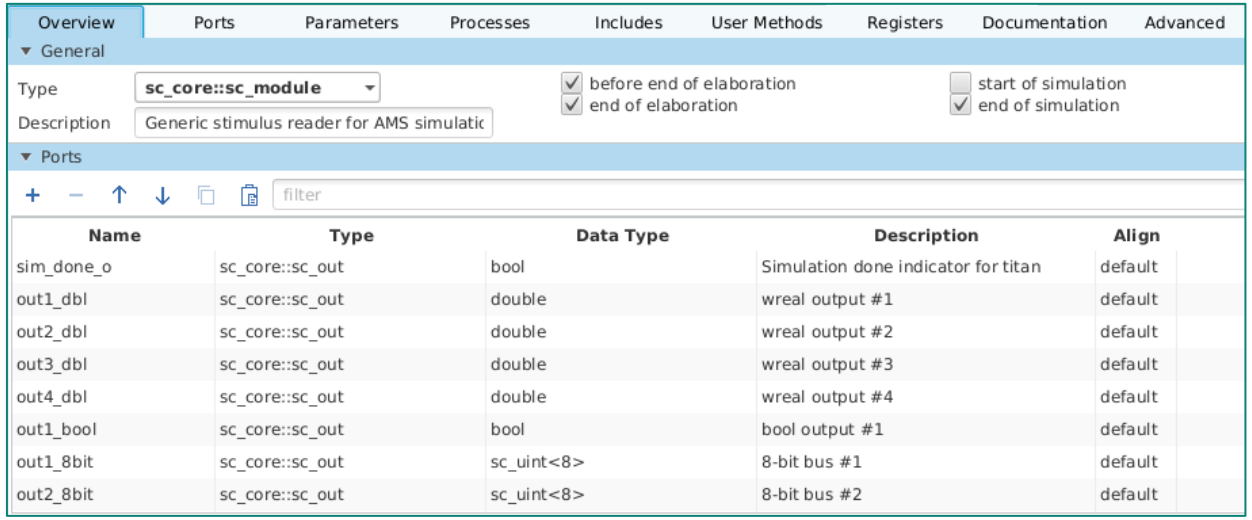


Fixed ports:

- Interface to register bank (Can be connected e.g. to I2C master model)
- Interface to Password Generator (Can be connected to custom pwd generator model)
- *Any change to these ports requires manual C++ code adjustment*

Customizable ports:

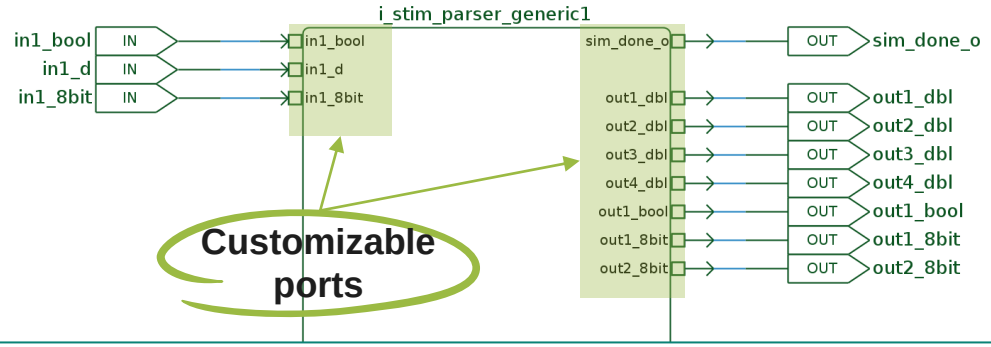
- Three different port types:
  - Bool → can only be set to ‘0’ or ‘1’
  - 8-bit busses → Can be set through hex codes (e.g. 0xAB)
  - Double → double precision value (will be translated in a wreal signal when exported to virtuoso)
- **Port type can be changed from GUI (no code changes needed)**



Name	Type	Data Type	Description	Align
sim_done_o	sc_core::sc_out	bool	Simulation done indicator for titan	default
out1_dbl	sc_core::sc_out	double	wreal output #1	default
out2_dbl	sc_core::sc_out	double	wreal output #2	default
out3_dbl	sc_core::sc_out	double	wreal output #3	default
out4_dbl	sc_core::sc_out	double	wreal output #4	default
out1_bool	sc_core::sc_out	bool	bool output #1	default
out1_8bit	sc_core::sc_out	sc_uint<8>	8-bit bus #1	default
out2_8bit	sc_core::sc_out	sc_uint<8>	8-bit bus #2	default

Table of contents

Generic Stimulus Parser – Advantages




```
int stim_parser_generic::parse_line(std::string com,
 std::string args){
    int ret=0;
    if(com.compare(0,4,"SET_")==0){
            ret=set_voltage(com, args);
    }else if(com.compare(0,5,"RAMP_")==0){
            ret = ramp_voltage(com,args);
    }else if(com.compare(0,6,"CHECK_")==0){
            ret = check_voltage(com,args);
    ...
    }else {
        ret=-2;
        STREAM_REPORT_INFO_VERB(MSGID, "Line not valid", verbosity);
    }
    return ret;
}
```

The three **highlighted** functions are coded to dynamically select the port to be written or read in run time:

- **Same code regardless of the number of ports**
- **No need to re-verify the IP whenever its interface changes**

Scalability:

- Possibility to **add/remove/rename** ports **from GUI** without changing C++ code
- This makes the block **suitable** for **DUTs of any size**

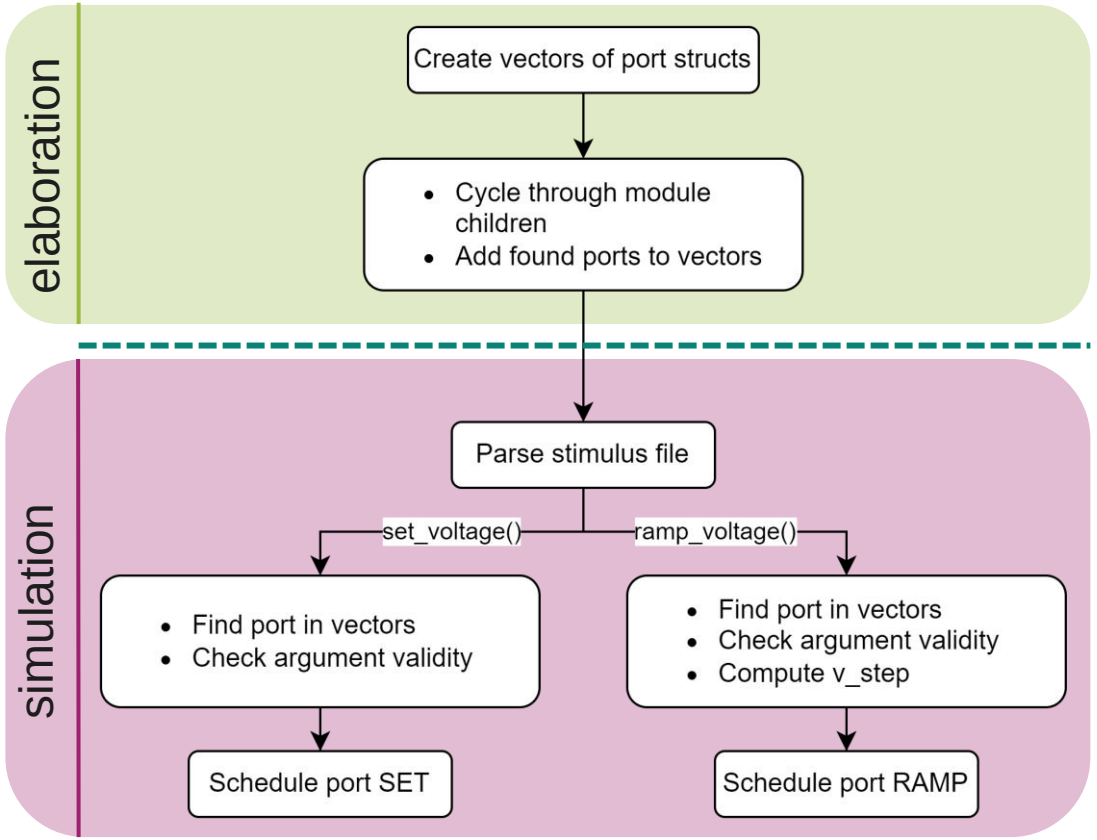
Flexibility:

- Customizable ports can be of three different types:
  - **Bool** → can only be set to '0' or '1'
  - **8-bit** busses → Can be set through hex codes (e.g. 0xAB)
  - **Double** → double precision value (will be translated in a wreal signal when exported to virtuoso)
- Their port type can be changed from GUI (no code changes needed)
- This makes the block **suitable** for **both digital and AMS** simulations

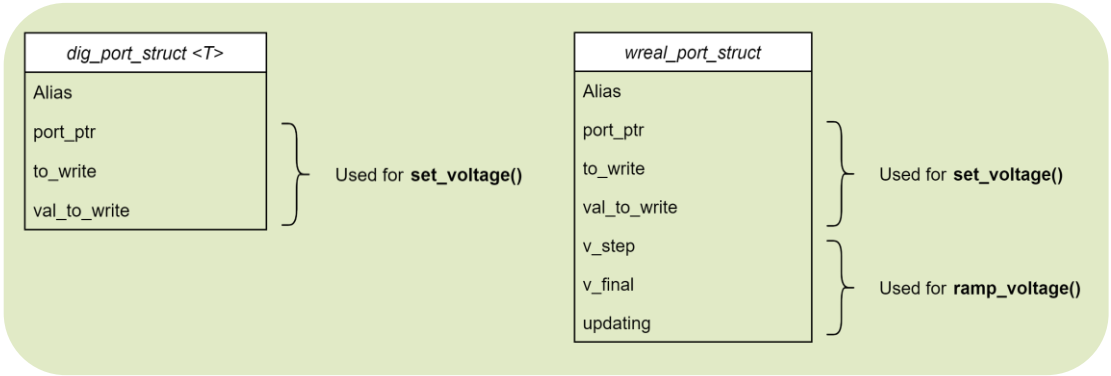
Concurrency:

- **Commands** can be **executed** in **parallel**
  - E.g. you can ramp two supply signals at the same time or stimulate a digital pin while an analog port is still ramping.

Dynamic port selection

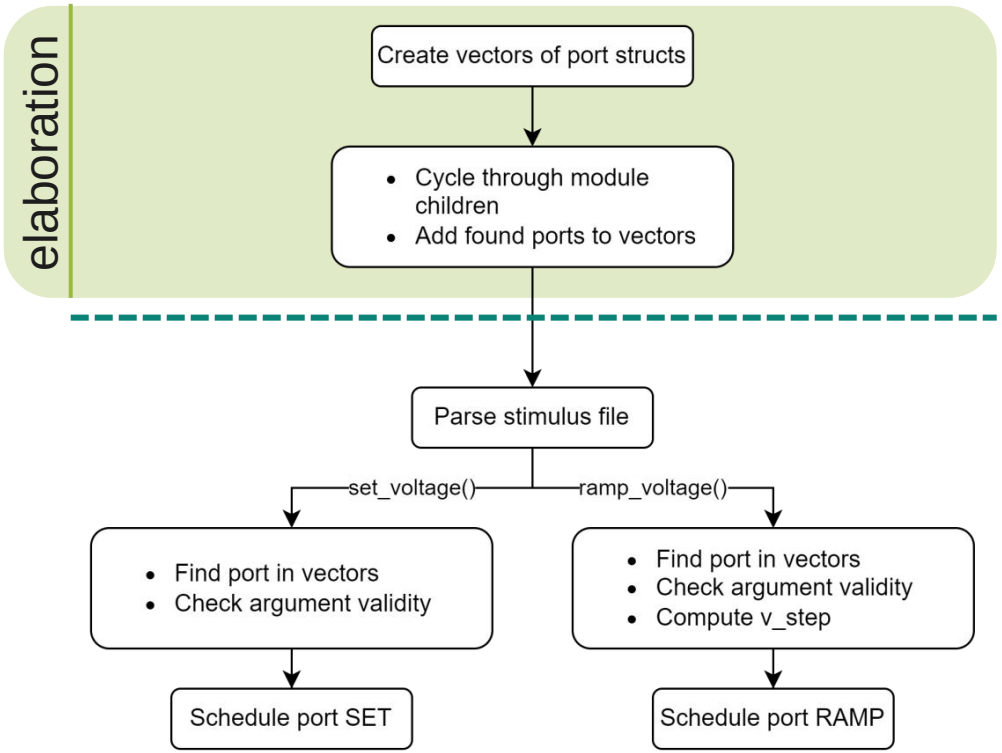


Dynamic port selection (elaboration phase)

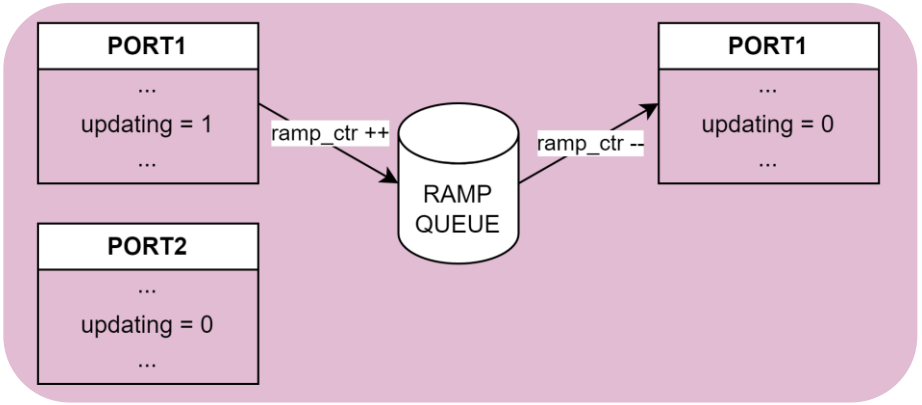


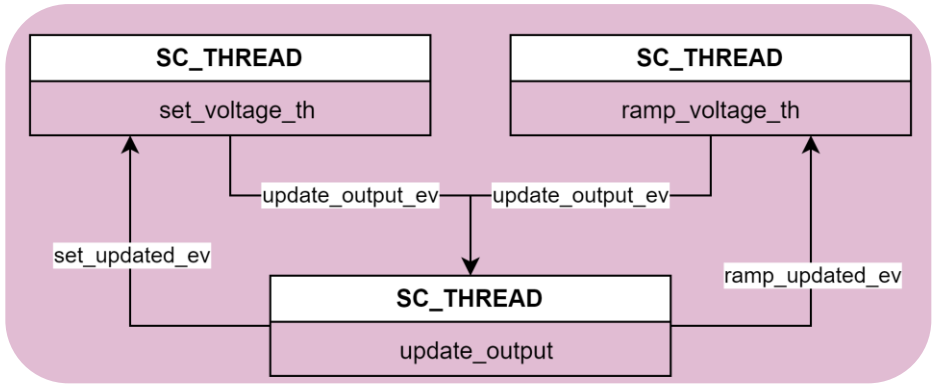

```
std::vector<digital_port_struct<sc_core::sc_in<bool>>> bool_inputs;
std::vector<digital_port_struct<sc_core::sc_in<sc_dt::sc_uint<8>>>> uint_inputs;
std::vector<wreal_port_struct<sc_core::sc_in<double>>> wreal_inputs;
```

```
for (const auto &child : this->get_child_objects()){
    sc_core::sc_out<double> * o_dbl = dynamic_cast<sc_core::sc_out<double> *>(child);
...
    if (o_dbl != nullptr) {
        wreal_port_struct<sc_core::sc_out<double>> tmp_wreal_struct (o_dbl->basename(),o_dbl);
        std::cout << "Found output port: " << child->basename() << " of type DOUBLE" << std::endl;
        s.wreal_outputs.push_back(tmp_wreal_struct);
     }
...
}
```



Dynamic port selection (simulation phase)





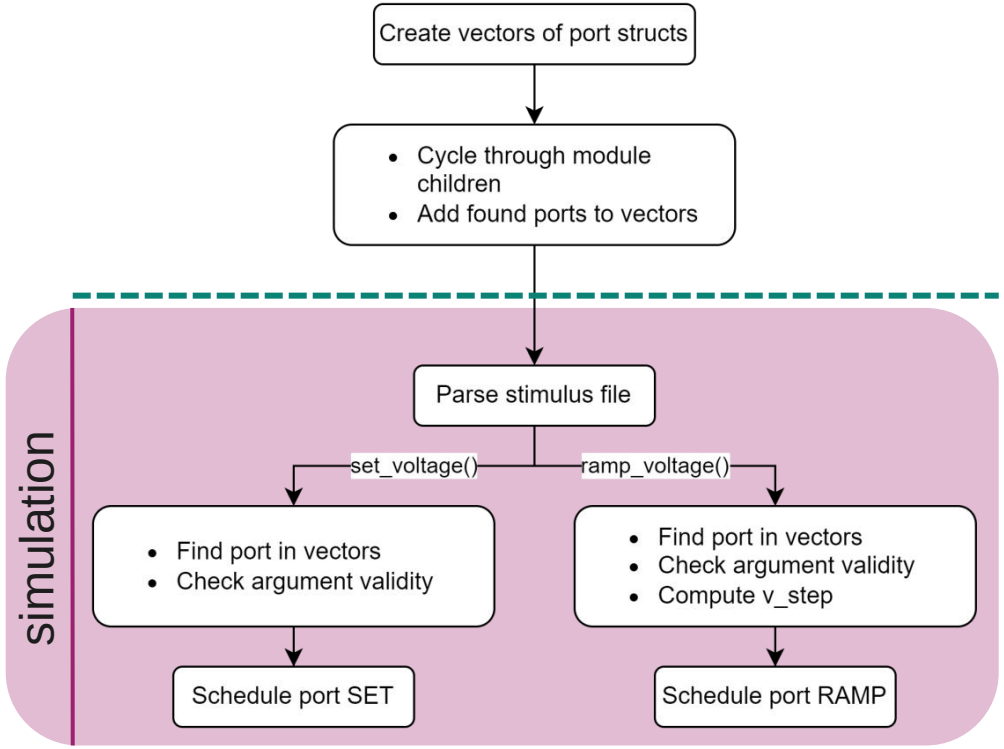


Table of contents

Generic Stimulus Reader – Features

Comments:

- # Line Comment... #
- ## Multi line comment
  continues here ##

Commands for double/wreal outputs:

- Ramp port "PORT_NAME" from current voltage to "V_FINAL" in "T_DURATION_US" µs.
  RAMP command does not block the execution of the script.
  - **RAMP_**<PORT_NAME> <V_FINAL> <T_DURATION_US>
- Immediately set wreal/double output port "PORT_NAME" to "V_FINAL“
  - **SET_**<PORT_NAME> <V_FINAL>

Commands for Boolean and 8-bit outputs:

- Set the 8-bit port "PORT_NAME" to <VALUE>
  - **SET_**<PORT_NAME> <VALUE>

Commands for input ports:

- Check the value of an **input** port against an expected value:
  - **CHECK_**<PORT_NAME> <EXPECTED_VALUE>

Commands for PW IF:

- Send “PW” as a password through the password generator
  - **SEND_PASSWORD** <PW>

Commands for Reg. Bank IF:

- Perform read operation on the Register Bank IF (**e.g. I2C**), and check if read data is equal to EXP_DATA.
  - **I2C_READ** <ADDR> <EXP_DATA>
- Perform write operation on the Register Bank IF (**e.g. I2C**)
  - **I2C_WRITE** <ADDR> <DATA>

Wait statements:

- Wait for a specific amount of time expressed in MS, US or NS before executing the next command
  - **WAIT_<MS/US/NS>** <TIME>
- Wait for a given **boolean** signal to change
  - **WAIT_POSEDGE** <BOOL_IN_PORT> or
  - **WAIT_NEGEDGE** <BOOL_IN_PORT> or
  - **WAIT_CHANGE** <BOOL_IN_PORT>
- Wait for all the executing ramps to end before proceeding with the next commands
  - **WAIT_RAMPS**

ALIAS:

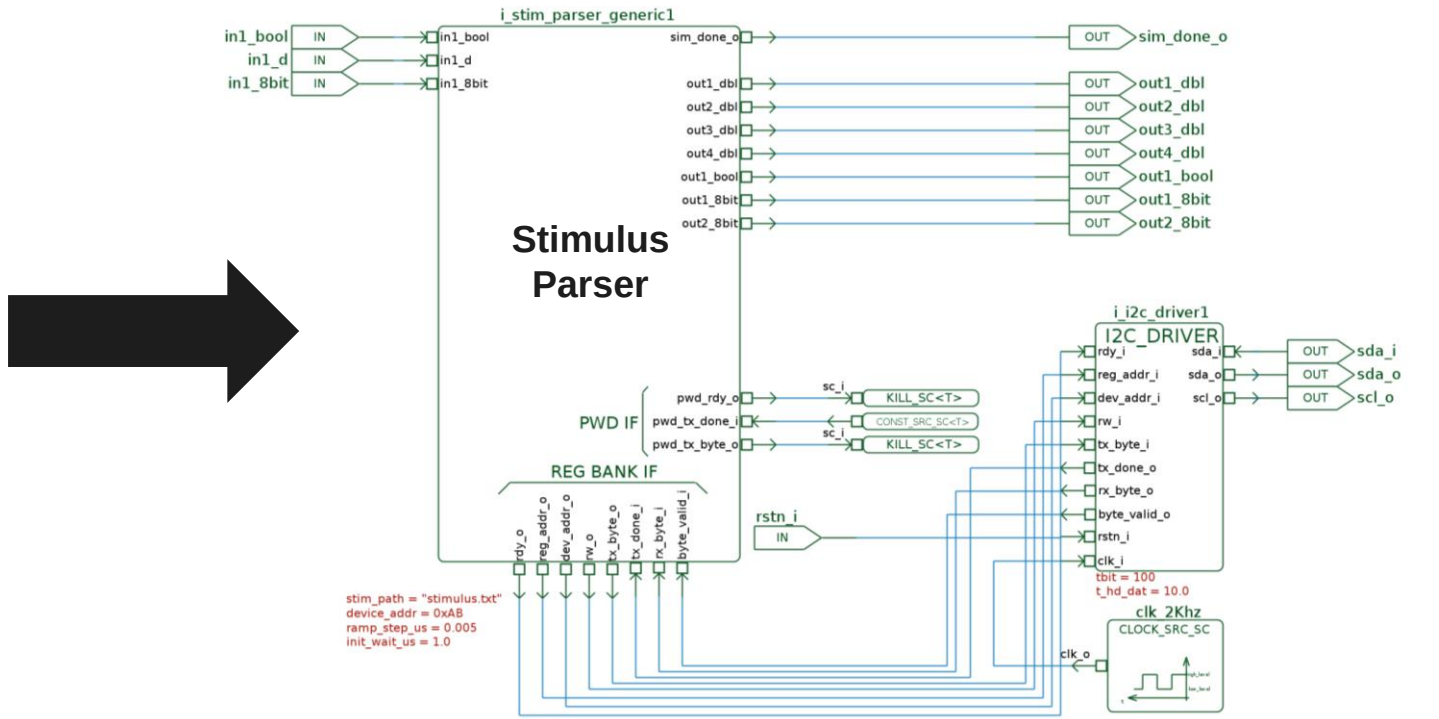
- Create an alias for a given input or output port
  - **ALIAS** <PORT_NAME> <PORT_ALIAS>

Table of contents

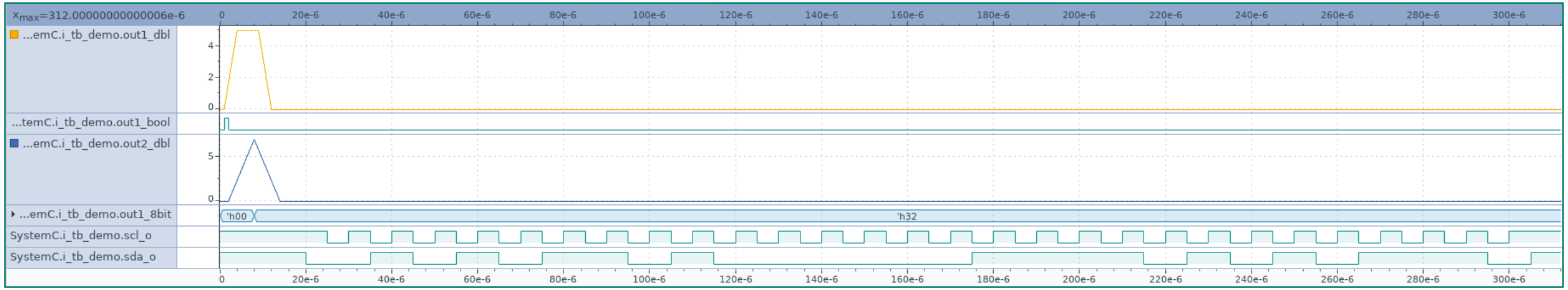
Generic Stimulus Reader – Coside simulation

```
STIM

RAMP_OUT1_DBL 5 3
SET_OUT1_BOOL 1
WAIT_US 1
SET_OUT1_BOOL 0
RAMP_OUT2_DBL 7 6
WAIT_RAMPS
RAMP_OUT2_DBL 0 6
SET_OUT1_8BIT 0x32
WAIT_US 1
RAMP_OUT1_DBL 0 3
I2C_WRITE 0x03 0xAB
WAIT_US 1

EXIT
```

Stimulus file



Waveforms



Questions & answers

Contact me at:

alessandro.putorti@infineon.com


Infineon