Describe of CRTG

This program works with GDB data bases which contains fields of data for generation of System Verilog constrained-random (CR) test module.

Before start…

Example of CRTG shell:
- sv (folder)
- db (folder)
- crtg.exe
- help.pdf

Field description

Common

- «Test name»
- «CR vectors amount»: amount of CR-vectors in array;

Create, Open, Save, Close:

Actions with the databases.

External plug-in file (in «Symbol defines» menu item):

external file of definitions, for example:

```
defines.v
```

Internal definitions:

definitions which will be placed inside of the test module, for example:

```
PATTERNS_FILE_PATH "sv/timers/rand/patterns/tc2/"
```

Connect to the DUT:

Module name, instance name, internal, external signals and their width.

Comment

Constrained random signals:

CR-signals which supplies to the DUT.

- «Name of signal»
- «Width» (about values of it field see *this*)
- «connect to the module»: this signal will be connected to the DUT;
- «delay relative to clock»: if checked - 1ns, if not – no delay;
- «Range»: range of random values;

• «Value(s)»: values which CR-signal can take and their weight (i.e. probability of finding a signal in this state)

• «Dependence of value from another signal(s) value(s)»: describe a signal name(first field) from value of which(second field) depends what value will have definable signal(third field) or which signal will be assigned to it(forth field).

Direct signals:

direct signals which supplies to the DUT.

• «Name of signal»
• «Width» (about values of it field see *this*)
• «Permanent» - permanent value of signal, for example:

```
CLR
```

• «Periodic» periodic value of signal.

«Init value»: initial state of the signal, for example:

```
SET
```

«Half-period time, ns»:

```
15 (or HP_CLK_DELAY)
```

• «Arbitrary»: arbitrary values of signal.

«Initial value»
«Initial delay»: delay of initial state for N positive edges of clock, for example:

```
20 (or INIT_DELAY)
```

«State»: state of signal
«State`s delay»: delay (for N positive edges of clock) of above-mentioned state of signal. If state of signal should be unchanged up to the end of simulation, this field should be empty.

How work with program

Firstly need create the database. Press «Create database» button or selest «Create new» item in «DataBase» menu. Then need fill above-mentioned fields (which need). After it done need save the database – press «Save database» button or select «Save» item in «DataBase» menu for it. Then need define of System verilog file name – press «…» button for it. And then press «Generate test» button or select «Generate» item in the «Run» menu.

If a database exists, need press «Open database» button or select «Open» item in the «DataBase» menu.

Important notes

1. In file of definitions must be defined:

    CONSTRAINED_RANDOM_GENERATING ***or***
    CONSTRAINED_RANDOM_FROM_PATTERN parameters (without values).

If you defined first parameter input *generated* CR-signals and responses from the DUT will be dumped to the file of pattern signals <pat_[name_test].txt**>**

If you defined second parameter input CR-signals and responses from the DUT will be read from the pattern file. CR-signals will be assigned into the DUT, responses from the DUT will be compared with the data from the pattern file.

2. In file of definitions must be defined path to the patterns file (see this)

3. The data base after work with it should be closed.

4. Signal (direct or constrained-random) must be saved after any modification.

5. Information must be saved into the database after any modifications of data.

6. When need copy or insert item(s) of data need select it from *up-left* item to *down-right* item.

7. Selected data is not highlighted.

8. Actions of menu items and buttons allowed only on necessity .

9. After a database was copied, a current database must be closed if need edit a new database.

Definitions of values

Symbol definitions examples:

- SET
- CLR
- ADDR_WIDTH

Bit capacity for «Width» fields:

If field is empty then signal has 1-bit width.

Values:
- decimal, for example: 2, 8 etc.,
- symbol definition, for example: ADDR_WIDTH, DATA_WIDTH*2

Format hexadecimal values:

- 1-bit: 1, 0, 1'b0, 1'b1, 1'bz
- 2-bit: 2'b01, 2'bz0
- 8-bit: 'h23
- 16-bit: 'h1234