SystemC – Times (07A)

SystemC

Young Won Lim
05/28/2012

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Young Won Lim
05/28/2012

Based on the following original work

[1] Aleksandar Milenkovic, 2002
CPE 626 The SystemC Language – VHDL, Verilog Designer's Guide
http://www.ece.uah.edu/~milenka/ce626-02S/lectures/cpe626-SystemC-L2.ppt

[2] Alexander de Graaf, EEMCS/ME/CAS, 2010
SystemC: an overview ET 4351
ens.ewi.tudelft.nl/Education/courses/et4351/SystemC-2010v1.pdf

[3] Joachim Gerlach, 2001
System-on-Chip Design with Systent of Computer Engineering
http://www2.cs.uni-paderborn.de/cs/ag-hardt/Forschung/Data/SystemC-Tutorial.pdf

[4] Martino Ruggiero, 2008
SystemC
polimage.polito.it/~lavagno/codes/SystemC_Lezione.pdf

[5] Deepak Kumar Tal, 1998-2012
SystemC Tutorial
http://www.asic-world.com/systemc/index.html

SC_TIME_UNIT

SC_FS = 0
SC_PS = 1
SC_NS = 2
SC_US = 3
SC_MS = 4
SC_SEC= 5

SC_TIME_UNIT

SC_FS = 0
SC_PS = 1
SC_NS = 2
SC_US = 3
SC_MS = 4
SC_SEC= 5

sc_set_time_resolution()
sc_get_time_resolution()

SC_ZERO_TIME

SC_ZERO_TIME represents 0 time delay

but may involve several delta simulation time.

use this constant whenever creating a delta notification or a delta time-out.

SC_TIME

represents simulation time and time intervals, including delays and time-outs.

```
sc_set_time_resolution (1, SC_PS);

sc_time         t1 (1, SC_NS);
sc_time         t2 (2, SC_PS);
sc_time         t3, t4 (3, SC_PS), t5 (4, SC_PS);
```

t1 → 1 pico second
t2 → 2 pico second
t4 → 3 pico second
t5 → 4 pico second

SC_CLOCK

Clocks in modeling low-level hardware
Clocks incurs many events
    updating event-> clocks would involve siginificant simulation overhead

sc_clock : a built-in hierarchical channel
    generating timing signals to synchronize event
    supporting multiple clocks with arbitrary phase

sc_clock clock_name (“clock_label”, period [, duty_ratio, offset, initial_value]);
        sc_clock clock1 ("clock1", 20, 0.5, 2, true);

Simulation Control

sc_clock my_clock (“CLK”, 20, 0.5);

Simulation Start : from the top-level function sc_main()
sc_start() / sc_start(n)

Simulation Stop : from within any process
sc_stop()

Advanced simulation control: self-made clock
sc_initialize()
sc_cycle(n)

```
sc_clock my_clock (“CLK”, 20, 0.5 );
sc_start(200);
```

```
sc_initialize();
for (int i=0; i<=200; i++) {
        clock = 1;
        sc_cycle(10);
        clock = 0;
        sc_cycle(10);
}
```

Simulation Scheduler

Step 1: All clock signals to be changed are assigned their new value.

Step 2: All SC_METHOD / SC_THREAD processes
with changed inputs are executed.
SC_METHOD: The entire bodies are executed
SC_THREAD: executed until the next wait()
(The execution order of inter processes cannot be determined)

Step 3: All **outputs** of SC_CTHREAD processes that are triggered
are **updated** and **saved** in a queue to be executed in step 5.
All **outputs** of SC_METHOD / SC_THREAD processes
that were executed in step 1 are also **updated**.

Step 4: Step 2 and Step 3 are repeated until *no signal changes* it's value.

Step 5: All SC_CTHREAD processes that were triggered and queued in
step 3 are **executed**. (Non-deterministic execution order)
Their outputs are updated at the next active edge
(when step 3 is executed), and therefore are saved internally.

Step 6: **Simulation time** is advanced to the next clock edge and the
scheduler goes back to step 1

References

[1] Aleksandar Milenkovic, 2002
CPE 626 The SystemC Language – VHDL, Verilog Designer's Guide
http://www.ece.uah.edu/~milenka/ce626-02S/lectures/cpe626-SystemC-L2.ppt

[2] Alexander de Graaf, EEMCS/ME/CAS, 2010
SystemC: an overview ET 4351
ens.ewi.tudelft.nl/Education/courses/et4351/SystemC-2010v1.pdf

[3] Joachim Gerlach, 2001
System-on-Chip Design with Systent of Computer Engineering
http://www2.cs.uni-paderborn.de/cs/ag-hardt/Forschung/Data/SystemC-Tutorial.pdf

[4] Martino Ruggiero, 2008
SystemC
polimage.polito.it/~lavagno/codes/SystemC_Lezione.pdf

[5] Deepak Kumar Tal, 1998-2012
SystemC Tutorial
http://www.asic-world.com/systemc/index.html