

VOORBLAD SCHRIFTELIJKE TOETSEN

MINOR	: Embedded Systems
TOETSCODE	: HM-ES-sc1
GROEP	: D-MNES
TOETSDATUM	: 18 januari 2011
TIJD	: 6^E/7^E uur (13.00-14.00 uur)
AANTAL PAGINA'S (incl. dit voorblad)	: 5
DEZE TOETS BESTAAT UIT	: 3 open vragen (aantal) 0 MC-vragen (aantal)
GEBRUIK HULPMIDDELEN	: JA
TOEGESTANE HULPMIDDELEN	: Tijdens dit tentamen mogen alle boeken, dictaten, aantekeningen enz. worden gebruikt.
OVERIGE OPMERKINGEN	: Geen
OPSTELLER VAN DEZE TOETS	: Harry Broeders
NAAM 2^E LEZER	: Geen

KLAS(SEN) : D-MNES (Minor Embedded Systems)	BLAD : 1 van 4 BLADEN
TOETS : Hardware/Software Codesign with SystemC	DOCENT : Harry Broeders
CODE : HM-ES-sc1	DATUM : 18 januari 2011
KWARTAAL: 2	TYPE : tentamen
	TIJD : 13.00 – 14:30

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

Bij elke deelopgave staat tussen haakjes het maximale aantal te behalen punten vermeld.
Eindcijfer = (aantal behaalde punten + 10) / 10.

1. Gegeven is het volgende SystemC programma:

```
#include <systemc>
using namespace sc_core;
using namespace sc_dt;
using namespace std;

SC_MODULE(ClkDiv2) {
    sc_in_clk c_in;
    sc_in<bool> reset;
    sc_out<bool> c_out;
    SC_CTOR(ClkDiv2) {
        SC_METHOD(run);
        sensitive << c_in.pos() << reset;
    }
private:
    void run() {
        if (reset.read())
            c_out.write(true);
        else
            if (c_in.posedge())
                // inverteer de uitgang met behulp van !
                c_out.write(!c_out.read());
    }
};

SC_MODULE(TB) {
    sc_clock c_in;
    sc_signal<bool> c_out;
    sc_signal<bool> reset;
    /* deze constructor moet jij schrijven! */
private:
    void run() {
        /* deze code moet jij schrijven! */
    }
    ClkDiv2 dut;
};

int sc_main(int argc, char* argv[]) {
    TB tb("tb");

    sc_trace_file *tf(sc_create_vcd_trace_file("trace"));
    tf->set_time_unit(1, SC_NS);
    sc_trace(tf, tb.c_in, "c_in");
    sc_trace(tf, tb.c_out, "c_out");
    sc_trace(tf, tb.reset, "reset");
}
```

Zie volgende blad ⇨

KLAS(SEN) : D-MNES (Minor Embedded Systems)	BLAD : 2 van 4 BLADEN
TOETS : Hardware/Software Codesign with SystemC	DOCENT : Harry Broeders
CODE : HM-ES-sc1	DATUM : 18 januari 2011
KWARTAAL: 2	TYPE : tentamen TIJD : 13.00 – 14:30

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

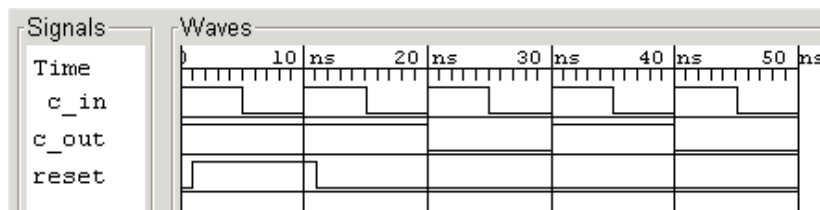
```

sc_start(51, SC_NS);

sc_close_vcd_trace_file(tf);
return 0;
}

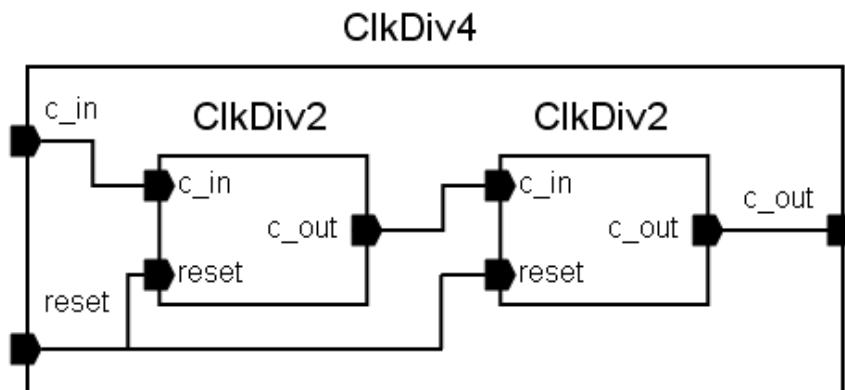
```

Het uitvoeren van dit programma moet de volgende waveform opleveren:



- A. (20) Geef de code van de constructor van de testbench TB en de code van de memberfunctie TB::run zodat het programma de bovenstaande waveform oplevert.

Door twee ClkDiv2 modulen met elkaar te verbinden volgens onderstaand schema kan een ClkDiv4 module gemaakt worden.



- B. (20) Geef de complete SystemC code van de module ClkDiv4.
- C. (5) Hoe moet de test bench module TB worden aangepast zodat deze voor het testen van de module ClkDiv4 gebruikt kan worden. Geef alleen de regel(s) die vervangen moet(en) worden en geef ook de aangepaste code.
- D. (10) Teken de waveform van de c_in, c_out en reset signalen. Ga ervan uit dat je vraag B en C correct hebt beantwoord.

KLAS(SEN) : D-MNES (Minor Embedded Systems)	BLAD : 3 van 4 BLADEN
TOETS : Hardware/Software Codesign with SystemC	DOCENT : Harry Broeders
CODE : HM-ES-sc1	DATUM : 18 januari 2011
KWARTAAL: 2	TYPE : tentamen
	TIJD : 13.00 – 14:30

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

2. Gegeven is het volgende SystemC module:

```
SC_MODULE(Fac) {
    sc_in_clk clk;
    sc_in<bool> go_i;
    sc_in<sc_uint<4> > n_i;
    sc_out<bool> done_o;
    sc_out<sc_uint<41> > f_o;
    SC_CTOR(Fac) {
        SC_THREAD(run);
        sensitive << clk.pos();
    }
private:
    void run() {
        wait();
        while(1) {
            do {
                wait();
            } while (!go_i.read());
            sc_uint<4> n = n_i.read();
            sc_uint<41> f = 1;
            wait();
            while (go_i.read() && n != 0) {
                f = f * n;
                n = n - 1;
                wait();
            }
            if (go_i.read()) {
                f_o.write(f);
                done_o.write(true);
            }
            do {
                wait();
            } while (go_i.read());
            done_o.write(false);
        }
    }
};
```

Deze module berekent fac (n), oftewel n!. Het model is een zogenoemd cycle-accurate model. Uit testen is gebleken dat dit model correct werkt.

- A. (10) Hoeveel clock cycles duurt het volgens bovenstaande model om 15! te berekenen? Bereken het aantal clock cycles vanaf het hoog worden van go_i tot het hoog worden van done_i. **Verklaar je antwoord!**

Zie volgende blad ⇨

KLAS(SEN) : D-MNES (Minor Embedded Systems)	BLAD : 4 van 4 BLADEN
TOETS : Hardware/Software Codesign with SystemC	DOCENT : Harry Broeders
CODE : HM-ES-sc1	DATUM : 18 januari 2011
KWARTAAL : 2	TYPE : tentamen
	TIJD : 13.00 – 14:30

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

- B. (15) We willen vervolgens een RTL model van deze module maken bestaande uit een datapath en een controller (zoals op het college behandeld). Geef het **blokschema** van het **datapath** op RTL niveau. Maak gebruik van de volgende RTL bouwstenen: register, multiplexer, multiplier, subtractor en comparator. Geef ook duidelijk aan welke data- en statussignalen door het datapath gegenereerd worden en welke data- en controlsignalen door het datapath gebruikt worden.
- 3.** In TLM 2.0 worden twee modelstijlen gedefinieerd: Loosely Timed (LT) en Approximated Time (AT). Een LT model kan sneller gesimuleerd worden dan een AT model.
- A. (10) Geef de belangrijkste redenen waarom een LT model sneller gesimuleerd kan worden dan een AT model.