NÉV: Vasvári Olivér

IT Eszközök Technológiája 3. házi feladat

Kiadva: 2017-09-25 Beadási határidő: 2017-10-02 12h Beadható: 2017-10-06 12h

A házi feladatot a tantárgyi portálon kell beadni, a beadási határidőig. A beadási határidő után még néhány napig a házi feladat beadható, ennek lejárta után viszont semmilyen indokkal nem fogadható el. Csak az eredményt és a nevet kell felírni (lehetőség szerint elektronikusan, mivel a feltöltés maximális mérete 2MB), a levezetések nem szükségesek.

1. Olvassa el a következő cikket!
   A. Takach, "High-Level Synthesis: Status, Trends, and Future Directions," in IEEE Design & Test, vol. 33, no. 3, pp. 116-124, June 2016.

   A cikket elolvastam

2. A logikai szintézis melyik módszerét magyarázza Arató professzor az index cikkét illusztráló képen?
   Készítse el a hétszegmenses kijelző valamelyik szegmensének kifejezését ezzel a módszerrel!

   K-Tábla
   Középső szegmens:

Szám	Engedélyezett
0000	0
0001	0
0010	1
0011	1
0100	1
0101	1
0110	1
0111	0
1000	1
1001	1
…	X

	C=0, D=0	C=1, D=0	C=1, D=1	C=0, D=1
A=1, B=0	1	X	X	1
A=1, B=1	X	X	X	X
A=0, B=1	1	1	0	1
A=0, B=0	0	1	1	0

3. Készítse el egy aszinkron resettel rendelkező négybites BCD számláló SystemC modelljét a hozzátartozó tesztkörnyezettel együtt!
   A számláló legyen kaszkádosítható! Javasolt elnevezések: clock – órajel, reset – alacsony aktív reset, enable – magas aktív engedélyezés, q – kimenetek, carry – átvitel.
   (a forráskódot egyszerűen illessze bele ebbe a dokumentumba, vagy egy zip file-ba tömörítse össze és úgy töltse fel!)

```
SC_MODULE(counter) {
	sc_in_clk clock;
	sc_in<bool> reset;
	sc_in<bool> enable;
	sc_out<sc_uint<4> > q;
	sc_out<bool> carry;

	sc_uint<4> count;
	void work() {
		while (true) {
			wait();
			if (reset.read() == 1) {
				count = 0;
				q.write(count);
			}
			else if (enable.read() == 1) {
				if (count == 0) {
					carry = 0;
				}
				if (count == 9) {
					count = 0;
					carry = 1;
				}
				count = count + 1;
				q.write(count);
			}
		}
	}
	SC_CTOR(counter) {
		SC_THREAD(work);
		sensitive << clock.pos();
		count = 0;
	}
};

int sc_main(int argc, char **argv)
{
	// a "top" modul jelei
	sc_clock clk("CLK", 1.0, SC_NS);   // orajel generator

	sc_signal<bool> reset("RESET");
	sc_signal<bool> enable("ENABLE");
	sc_signal< sc_uint<4> > q("Q");
	sc_signal<bool> carry("CARRY");

	// a shiftregiszter peldanyositasa es bekotese
	counter cnt("COUNTER");
	cnt.clock(clk);
	cnt.reset(reset);
	cnt.enable(enable);
	cnt.carry(carry);
```

```
    // hullamforma kimenetet valasztunk. Megtekintheto pl. a gtkwave programmal.
    sc_trace_file *fp;
    fp = sc_create_vcd_trace_file("wave");
    sc_trace(fp, clk, "clk");
    sc_trace(fp, reset, "reset");
    sc_trace(fp, enable, "enable");

    // itt kezdodik a szimulacio
    // reset
    reset = true;
    sc_start(2, SC_NS);
    reset = false;
    sc_start(200, SC_PS);

    enable = true;
    sc_start(1, SC_NS);
    sc_start(100, SC_NS);

    // takaritas
    sc_close_vcd_trace_file(fp);

    return 0;
}
```