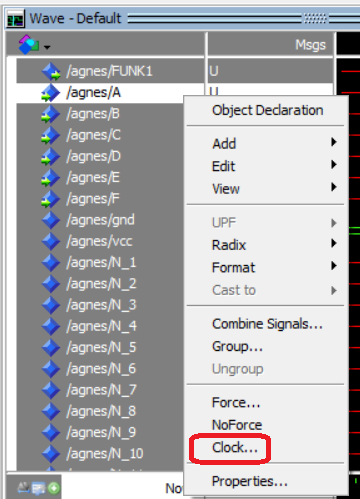
**Trigerių testavimas ModelSim**

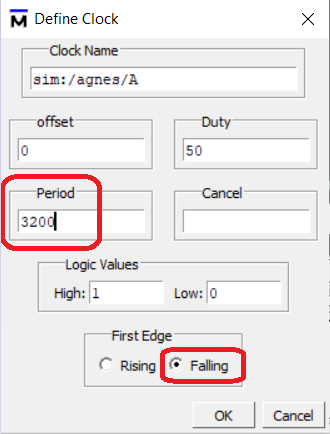
Modeliuodami (Lattice Diamond) sukurtų schemų darbą programa ModelSim, turime parengti testinius duomenis. Pateiksime kai kuriuos pasiūlymus, kurie palengvins šį darbą ir sutaupys laiko.

***Kombinacinės schemos testavimas***

Modeliuodami kombinacinės schemos darbą, testinius duomenis pirmąjį kartą suformuojame ModelSim programos atvertame ***Wave – Default*** lange pasirinkus kintamąjį ir kontekstiniame meniu pasirinkus **Clock:**



Atvertame**Define Clock** lange reikia nurodyti kintamojo, traktuojamo kaip sinchrosignalas, periodą ir pradinę reikšmę:



Taip padarius ***Transcript*** lange pamatysime tokią modeliavimo direktyvą:

**force -freeze sim:/agnes/A 0 0, 1 {1600 ps} –r 3200**

Patariame šias modeliavimo direktyvas išsaugoti tekstiniame faile. Tuomet kitą kartą modeliuojant schemą nereikės **Define Clock** lange nurodyti kintamųjų periodų ir pradinių reikšmių. Tiesiog po paskutinio  įterpsite nukopijuotas reikšmes ir po jų pridėsite direktyvą ***run 3200****.*

***Užduotos formos signalo formavimas***

Dabar parodysime, kaip reikia parengti panašias direktyvas, atitinkančias reikiamos formos įėjimo signalą **Sign**.

Tegul mums reikia suformuoti direktyvą, atitinkančią tokios formos signalą:

0 200 400 700

Tam panaudosime tokio pavidalo direktyvą:

**force -freeze sim:/<Projektas>/<Signalas> > <Pradinė reikšmė> <Prad.momentas, <Paskesnė reikšmė> <Paskesnės reikšmės pradžia> , <Paskesnė reikšmė> <Paskesnės reikšmės pradžia>, ...**

Mūsų formos signalui direktyva atrodys taip:

**force -freeze sim:/agnes/A 0 0, 1 200, 0 400, 1 700**

***Schemos su trigeriu testavimas***

Dabar paaiškinsime testo formavimą, panaudodami JK trigerio pavyzdį.

Tegul JK signalų reikšmių priklausomybę nuo kintamųjų **x2**, **x3**, **x4** aprašo tokia lentelė:

|  |  |  |
| --- | --- | --- |
| **x2** **x3** **x4** | ***K*** | ***J*** |
| 0 0 0 | 0 | 0 |
| 0 0 1 | 1 | 0 |
| 0 1 0 | 0 | 0 |
| 0 1 1 | 1 | 0 |
| 1 0 0 | 0 | 0 |
| 1 0 1 | 1 | 1 |
| 1 1 0 | 0 | 1 |
| 1 1 1 | 0 | 0 |

Testuojant statinį JK trigerį, **J** = **K** = 1 kombinacija negalima.

Iš lentelės matome, kad mūsų atveju JK trigeris nustatomas į būseną **Q** = 1, kai signalų **x2**, **x3**, **x4** reikšmių kombinacija lygi 1 1 0 (tuomet K = 0, J = 1), trigeris nustatomas į būseną **Q** = 0, kai signalų **x2**, **x3**, **x4** reikšmių kombinacija lygi 0 0 1 arba 0 1 1 (tuomet K = 1, J = 0), ir trigeris būsenos nekeičia, kai signalų **x2**, **x3**, **x4** reikšmių kombinacija lygi 0 0 0, 0 1 0, 1 0 0 arba 1 1 1 (tuomet K = 0, J = 0).

Testuojant reikia patikrinti, ar trigeris teisingai priima informaciją (0 ir 1) ir ją išsaugo. Tad testavimo seka bus tokia (pradinę būseną Q = 0 nustatysime **Reset** signalu):

|  |  |
| --- | --- |
| **Veiksmas** | **x2** **x3** **x4** |
| Išsaugome **Q** = 0 | 0 0 0 |
| Nustatome **Q =** 1 | 1 1 0 |
| Išsaugome **Q** = 1 | 0 1 0 |
| Nustatome **Q =** 0 | 0 0 1 |
| Išsaugome **Q** = 0 | 1 0 0 |
| Nustatome **Q =** 1 | 1 1 0 |
| Išsaugome **Q** = 1 | 1 1 1 |
| Nustatome **Q =** 0 | 0 1 1 |

Kaip matome, į trigerį įrašomas „1“, naudojant tą pačią **x2**, **x3**, **x4** reikšmių kombinaciją 1 1 0, į trigerį įrašomas „0“, naudojant abi **x2**, **x3**, **x4** reikšmių kombinacijas (0 0 1 ir 0 1 1). Informacijos saugojimui panaudojome visas 4 **x2**, **x3**, **x4** reikšmių kombinacijas.

**Atminkite**, kad pradžioje trigeris nustatomas į pradinę būseną **Q** = 0 **Reset** signalu, kai **C** = **0.** Derinant schemos darbą, kintamųjų **x2**, **x3**, **x4** reikšmės turi būti keičiamos taip pat tuomet, kai **C** = **0.**

Pagal lentelę taip turėtų atrodyti formuojamos signalų reikšmės:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Reset** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **X2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **X3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **X4**  0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800  P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Po diagrama pateikta laiko skalė, laikant, kad sinchronizacijos periodas lygus 100 ps.

Signalų formavimui panaudosime tokias modeliavimo direktyvas:

**force -freeze sim:/......./C 0 0, 1 {50 ps} -r 100**

**force -freeze sim:/......./Reset 0 0, 1 {25 ps}**

**force -freeze sim:/......./X2 0 0, 1 125, 0 225, 1 425, 0 725**

**force -freeze sim:/......./X3 0 0, 1 125, 0 325, 1 525**

**force -freeze sim:/......./X4 0 0, 1 325, 0 425, 1 625**

***Schemos su visais JK trigeriais testavimas***

Dabar paaiškinsime schemos su trijų tipų JK trigeriais testo formavimą. Tai naudinga norint pamatyti, kaip dinaminis trigeris „vartosi“, kai **J** = **K** = 1.

Kadangi statiniam ir dviejų pakopų trigeriams tokia signalų kombinacija (**J** = **K** = 1) negalima, galima panaudoti tokį principą: statinį ir dviejų pakopų trigerius sinchronizuojančią seką **C** nutrauksime prieš tą laiko momentą (tarkime, tai bus 600 ps), kai suformuojami signalai **J** = **K** = 1:

**force -freeze sim:/visi\_jk/C 0 0, 1 {50 ps} -r 100 -cancel 600,**

tuo tarpu dinaminį trigerį sinchronizuosime seka **Cdin:**

**force -freeze sim:/visi\_jk/Cdin 0 0, 1 {50 ps} -r 100.**

Grafiškai tai atitiks toks paveikslas:

**Statinis**

**Dviejų pakopų**

**Dinaminis**

**Rst**

**C**

**Cdin**

**J**

**K**

Modeliuodami tokios schemos darbą su direktyvomis:

**restart -f**

**force -freeze sim:/visi\_jk/Rst 0 0, 1 10**

**force -freeze sim:/visi\_jk/C 0 0, 1 {50 ps} -r 100 -cancel 600**

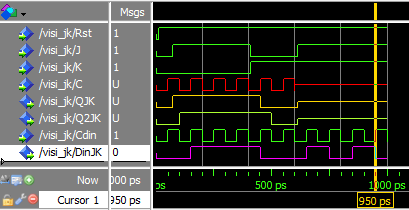
**force -freeze sim:/visi\_jk/Cdin 0 0, 1 {50 ps} -r 100**

**force -freeze sim:/visi\_jk/J 0 0, 1 70, 0 410, 1 600**

**force -freeze sim:/visi\_jk/K 0 0, 1 410**

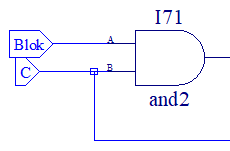
**run 1000**

gavome tokį vaizdą:



Matome, kad nutraukus **C** seką statinis (QJK) ir dviejų pakopų (Q2JK) trigeriai modeliavimo proceso nestabdo, o dinaminis trigeris (DinJK) „vartosi“, kai **J** = **K** = 1, o jo sinchronizavimas (Cdin) nestabdomas.

Galima tai pasiekti ir kitu būdu – blokuojant statinį ir dviejų pakopų trigerius sinchronizuojančią **C** seką prieš tą momentą, kai nustatomos reikšmės **J** = **K** = 1. Tam schemoje pridėsime IR elementą:



Statinio ir dviejų pakopų trigerių sinchronizavimui

Dinaminio trigerio sinchronizavimui

Modeliuodami tokios schemos darbą su tokiomis direktyvomis:

**restart -f**

**force -freeze sim:/visi\_jk\_1/Rst 0 0, 1 10**

**force -freeze sim:/visi\_jk\_1/C 0 0, 1 {50 ps} -r 100**

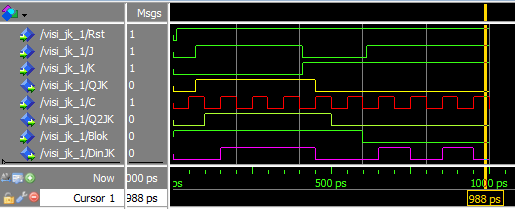
**force -freeze sim:/visi\_jk\_1/Blok 1 0, 0 {600 ps}**

**force -freeze sim:/visi\_jk\_1/J 0 0, 1 70, 0 410, 1 610**

**force -freeze sim:/visi\_jk\_1/K 0 0, 1 410**

**run 1000**

gavome tokį vaizdą:



Tad galima rinktis tą ar kitą variantą.