

2. Optimizacija algoritma

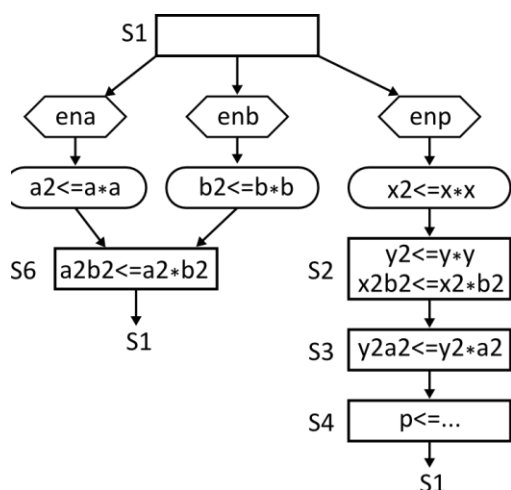
Vezje za izračun točk v elipsi bomo optimizirali, tako da bomo naredili časovno in prostorsko razdelitev operacij ter računski cevovod.

2.1 Sekvenčna izvedba algoritma

Algoritem za izračun točke v elipsi ima 4 vhodne parametre: a , b , x in y . V vezje bomo dodali uro in kontrolne signale:

- **ena** = '1', določa vnos parametra a
- **enb** = '1', določa vnos parametra b
- **enp** = '1', določa vnos koordinat (x, y) nove točke

Pri izvedbi algoritma bomo upoštevali, da parametra a in b nastavimo na začetku, potem pa naj se čim hitreje izvede množica ovrednotenij algoritma z različnima koordinatama x in y . Časovno zaporedje izračunov bomo opisali z algoritmičnim strojem (**Algorithmic State Machine**), ki ga predstavimo z diagramom:



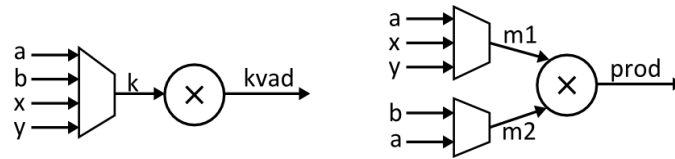
Legenda:

- S3 $y2a2 \leq y2 \cdot a2$ Stanje in stavek, ki se izvede v stanju
- enp Ovrednotenje pogoja $enp = '1'$
- $x2 \leq x \cdot x$ Stavek, ki se izvede ob pogoju

1. Naredi opis prehajanja stanj algoritmičnega stroja v jeziku VHDL. Definiraj notranji signal, ki shranjuje stanje in zapiši sekvenčni proces prehajanja stanj. V tem koraku še ne opisuj računskih stavkov, ki se izvajajo v stanjih ali ob pogojih.

Izvedbo algoritma smo razdelili na 5 stanj (S1 do S6), operacije pa se izvajajo tako, da potrebujemo v vsakem trenutku največ dva množilnika: kvadrirnik velikosti 8x8 bitov in množilnik 16x16 bitov. Ker jih bomo uporabili za izračun različnih izrazov, naredimo splošen kombinacijski opis množilnika in izbiralnika, ki določa vhodne signale.

2. Definiraj nepredznačene vektorje: 8-bitni k , 16-bitni $kvad$, $m1$, $m2$ in 32-bitni $prod$ ter zapiši izraze za izračun kvadrata in produkta. Opiši izbiralnik, ki pripelje na vhod kvadrirnika k signale a , b , x ali y , glede na stanje in aktiven vhodni kontrolni signal ena , enb ali enp . Opiši še izbiralnika na obeh vhodih drugega množilnika.

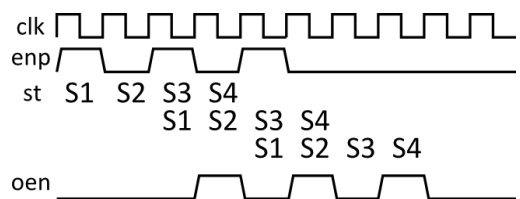


3. Dopolni sekvenčni proces prehajanja stanj z računskimi izrazi v katerih uporabi rezultate kombinacijskih množilnikov. Naredi sintezo vezja in zapiši zasedenost logičnih celic: _____

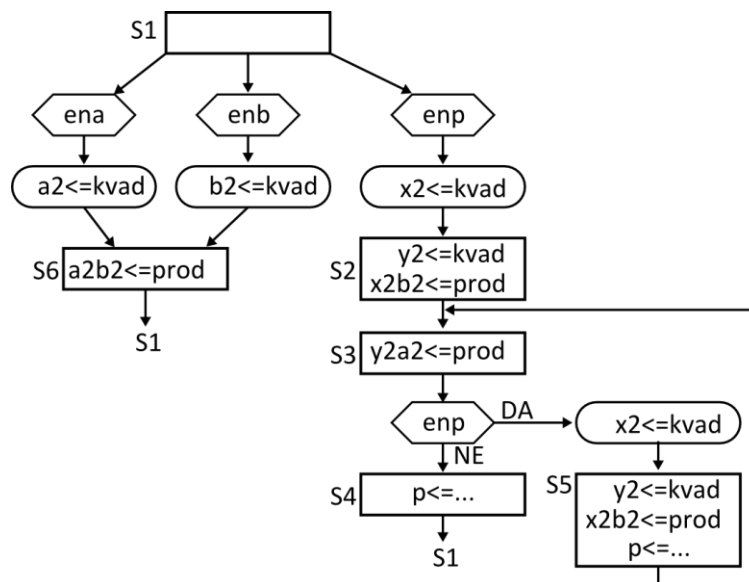
2.2 Cevovodna izvedba algoritma (*)

V sekvenčni izvedbi algoritma potrebujemo 4 zaporedna stanja (urne cikle), da izračunamo ali podana točka leži znotraj elipse. Paralelna izvedba algoritma izračuna rezultat v enem samem ciklu, vendar smo ugotovili da potrebuje precej strojnih virov za izvedbo operacij.

Računski cevovod je arhitektura, ki omogoča vzporedno izvajanje operacij na različnih računskih enotah. V vezju imamo dva množilnika, ki ju lahko uporabimo za časovno razporejanje operacij. V stanju S3 potrebujemo le drugi množilnik, zato se lahko izvede hkrati z novim stanjem S1, ki potrebuje kvadrirnik.



Za izvedbo cevovoda dodamo nekaj novih stavkov in dodatno stanje S5 ($S4+S2$), kot prikazuje digram stanj:



1. Naredi opis cevovodne izvedbe vezja in preizkusi delovanje na simulatorju. Dodaj v vezje izhod **oen**, ki se postavi na '1' vsakokrat, ko je na voljo nov izračunan rezultat (stanje S4 ali S5).