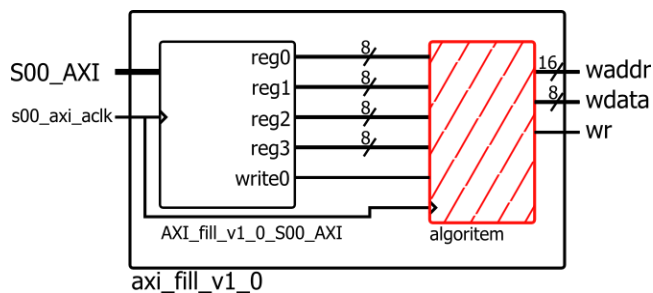


### 3. vaja: Algoritem pisanja v pomnilnik



Razvij digitalno vezje za algoritem, ki napolni kvadratno območje pomnilnika z izbrano vrednostjo. Pomnilnik velikosti 64k x 8 bitov je sestavljen in 256x256 8-bitnih vrednosti, ki predstavljajo točke slike.

- Naredi nov projekt v katerega vključi (Add design sources) komponento za prikazovanje slike VGArAm.vhd in algoritem.vhd ter izberi razvojno ploščo ZedBoard. Nato vključi v projekt še simulacijsko testno strukturo (Add Sources, Add simulation sources) tb\_vga\_algoritem.vhd.
  - VGArAm vsebuje signale za prikaz slike na izhodu VGA, signale za povezavo z blokovnim pomnilnikom in vmesnik za pisanje podatkov.
  - algoritem je ogrodje vezja v katerem opiši logiko za risanje kvadrata
  - tb\_vga\_algoritem vsebuje povezavo obeh komponent, model pomnilnika, določa testne vektorje za simulacijo ter naredi izhodno datoteko slika.txt
- Odpri **algoritem.vhd** in dopolni opis vezja, ki naj ob impulzu **write0='1'** začne postopek pisanja vrednosti v pomnilnik (nastavlja **waddr** in impulze na **wr**). Vezje naj zapiše vrednosti v kvadratno območje pomnilnika glede na nastavitve vhodnih registrov:
  - reg0**: določa barvo točke, ki se prenese na **wdata**
  - reg1**: določa velikost obarvanega kvadrata
  - reg2, reg3**: določata začetno točko (ogljische) kvadrata

Primer: reg0=15, reg1=3, reg2=1, reg3=2 naj povzroči, da se vrednost 15 zapiše v pomnilniške naslove: x0201, x0202, x0203 (prva vrstica), x0301, x0302, x0303 (druga), x0401, x0402, x0403
- Preveri vezje algoritem s simulacijo in naredi sintezo, nato pa vključi algoritem.vhd kot komponento v periferni vmesnik z vodilom AXI.
  - izberi: Tools > Create and Package NEW IP, Create a new AXI4 peripheral, Name: axi\_fill, na koncu izberi EDIT IP.
  - kopiraj algoritem.vhd v mapo axi\_fill\_1.0\hdl in datoteko vključi v projekt (Add source)
  - uredi notranjo komponento z vmesnikom AXI, tako da bo imela štiri 8-bitne izhode: reg0-reg3, ki jih poveži na slv\_reg0-3 in signal write0, ki gre na '1' ob pisanju v register slv\_reg0.
  - uredi zunanjo komponento axi\_fill, kopiraj spremembe priključkov notranje komponente, deklariraj notranje signale reg0-reg3 in write0 ter dodaj povezave v port map.
  - v axi\_fill dodaj izhodne priključke: waddr, wdata in wr, deklariraj komponento algoritem, ter jo vključi komponento s stavkom port map.
- Preveri strukturo vezja (Elaborated Design Schematic), naredi sintezo in ponovno pakiraj komponento axi\_fill.

Vir:

- Datoteke: [algoritem.zip](#)