

Dizajn: Sadrzi 8 memorijskih lokacija od 8 bita,

Ready: out std\_logic; Sistem je spreman za sledecu transakciju

Wr\_Data: in std\_logic\_vector (7 downto 0); Ulazni podatak

Rd\_Data: out std\_logic\_vector (7 downto 0); Izlazni podatak

Address: in std\_logic\_vector (2 downto 0); Adresa na koju se smesta ili sa koje se cita

Rd\_Cs: in std\_logic; Dozvola citanja

Wr\_Cs: in std\_logic; Dozvola upisa

op: in std\_logic\_vector (1 downto 0); Odabir operacije koja se vrsi nad ulazom Wr\_Data

"00" data\_to\_mem = not (wr\_data)

"01" logicko pomeranje za jedno mesto ulevo

"10" logicko pomeranje za jedno mesto udesno

"11" u memoriju se upisuje podatak bez menjanja (data\_to\_mem = wr\_data)

Test:

1. upisivati "setajucu" jedinicu na memorijsku lokaciju sa adresom 5. Testirati za oba smera.

Za 30% upisa proveriti memorijsku lokaciju

Kreirati sekvencu koja proverava operaciju "11" na memorijskim lokacijama sa adresom 0-3.

Sve transakcije moraju biti validne. (wr\_cs != rd\_cs)

1. kreirati sekvencu koja na memorijske lokacije sa parnom adresom upisuje slucajno generisane vrednosti sa slucajno generisanom operacijom a zatim proverava da li je vrednost uspesno upisana. Sve operacije moraju biti podjednako zastupljene. 20% transakcija treba da bude nevalidno (wr\_cs == rd\_cs)
2. Kreirati sekvencu u kojoj prvo treba upisati broj 3 na lokaciju 3 a zatim koristeci op i citanje obezbediti da na kraju na lokaciji 3 imamo upisanu vrednost 3\*4