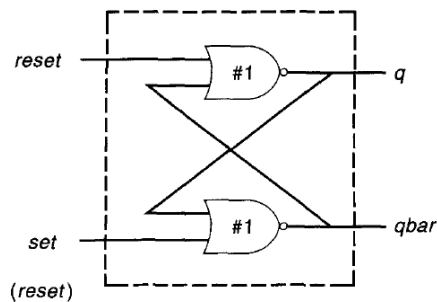


1. Kreirati vlastita kola (logičke kapije) po imenu *moj_and*, *moj_or* i *moj_invertor*, koristeći dvoulazne Verilog *nand* primitive. Provjeriti funkcionalnost ovih kola pomoću *stimulus* modula. (Napomena: podsjetiti se kako se realizuju proizvoljna logička kola pomoći NI kola).
2. Dvoulazno *xor* kolo se može napraviti od gore definisanih *moj_and*, *moj_or* i *moj_invertor* kola. Konstruisati *xor* modul koji realizuje logičku funkciju $z = x \bar{y} + \bar{x} y$. Ulazi su *x* i *y*, a izlaz je *z*. Napisati *stimulus* modul koji provjerava sve 4 kombinacije ulaznih signala.
3. Jednobitni potpuni sabirač se može zapisati i u obliku sume proizvoda:

$$sum = a \cdot b \cdot cin + \bar{a} \cdot b \cdot \bar{cin} + \bar{a} \cdot \bar{b} \cdot cin + a \cdot \bar{b} \cdot \bar{cin}$$

$$cout = a \cdot b + b \cdot cin + a \cdot cin$$

 Pretpostavljajući da su *a*, *b* i *cin* ulazi, a *sum* i *cout* izlazi, dizajnirati logičko kolo koje predstavlja jednobitni potpuni sabirač koristeći samo *and*, *not* i *or* logičke kapije. Napisati Verilog opis ovog kola. Dozvoljeno je koristiti Verilog primitive sa maksimalno 4 ulaza. Napisati *stimulus* modul koji će provjeravati funkcionalnost realizovanog jednobitnog potpunog sabirača za sve moguće ulazne kombinacije.
4. Logički dijagram RS latch-a sa kašnjenjem logičkih kapija je prikazan na slici 1. Napisati Verilog opis ovog kola, uključujući kašnjenje logičkih kapija. Napisati *stimulus* modul koristeći tabelu 1 i provjeriti izlaze.

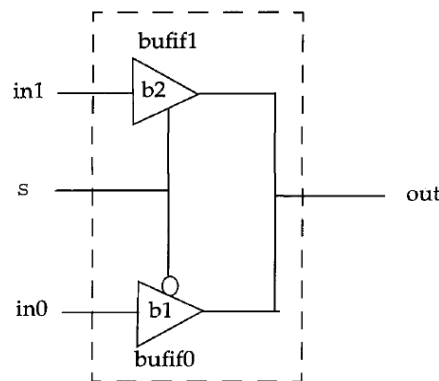


Slika 1

set	reset	q_{n+1}
0	0	q_n
0	1	0
1	0	1
1	1	?

Tabela 1

5. Realizovati MUX 2/1 koristeći *bufif0* i *bufif1* logičke kapije kao što je prikazano na slici 2. Kašnjenja za logičke kapije *b1* i *b2* su: *rise*=2, *fall*=4, *turn-off*=6. Primjeniti *stimulus* i provjeriti izlazne vrijednosti.



Slika 2