TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA ZAGREB

PROJEKTIRANJE I SIMULACIJA MEMORIJSKOG SUSTAVA

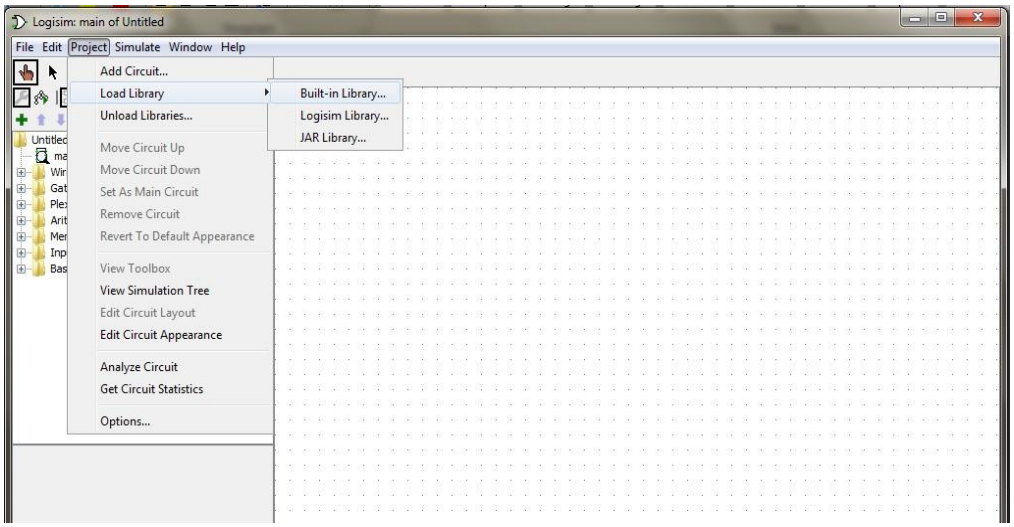
Uputa za laboratorijsku vježbu

Milan Korać, dipl.ing.

Zadaci:

1) **Pokrenuti program Logisim i otvoriti biblioteku s memorijskim modulima. Ispisati sve memorijske elemente koje sadrži ta biblioteka, nacrtati simbole i njihove pinove.**

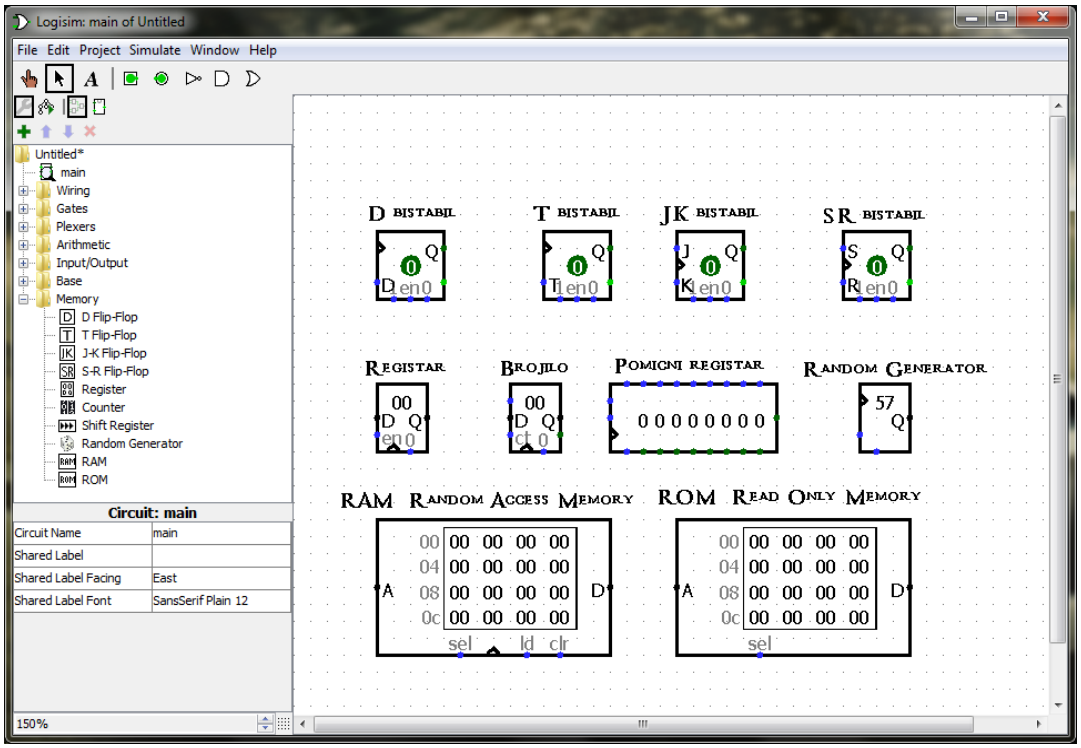
- Nakon što smo pokrenuli program Logisim, potrebno je dodati biblioteku koja sadrži memorijske module. Naziv te biblioteke je *Memory*.



*Navigacija:*Project→Load Library→Built-in Library→Memory→OK

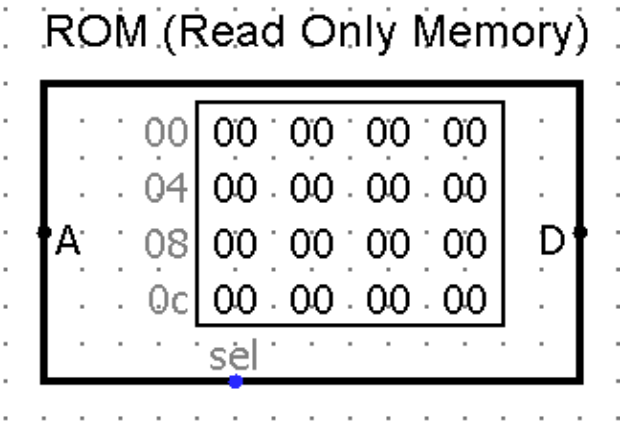
- Nakon što smo učitali biblioteku *Memory* otvaramo ju pritiskom na ikonu **+** pored mape *Memory*. Sadržaj biblioteke *Memory* čine: **D Flip-Flop, T Flip-Flop,J-K Flip-Flop, S-R Flip-Flop** (bistabili)**, Register, Shift Register, Random Generator, Counter**te **RAM** i **ROM** (memorijski moduli).
- Komponente na radnu površinu stavljamo pritiskom na lijevu tipku miša te pozicioniramo komponentu na radnu površinu programa te pritisnemo ponovno lijevu tipku kako bi postavili komponentu.

- Nakon što smo postavili komponente iz biblioteke *Memory*, postavljamo nazive sklopova (npr. RAM, T bistabil) tako da u alatnoj traci odaberemo ikonu  te pritiskom na radnu površinu omogućujemo upis određenog teksta. Program nudi i dodatne postavke tj. promjene teksta (vrsta fonta, veličina fonta) te poravnanje teksta (ulijevo, udesno, sredina, gore itd.). Postoji još jedan način kako možemo dodati naziv komponente, a to je da prilikom postavljanja komponete u tablici atributa u kategoriju ''*Label*'' napišemo naziv.

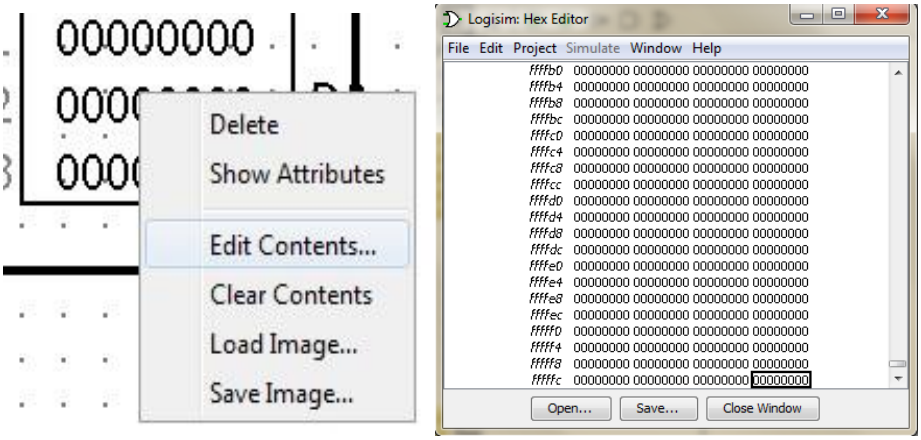


2) ''Postaviti na radnu površinu memorijski modul ROM i proanalizirati njegove ulaze i izlaze. Proučiti koliko memorijskih riječi može najviše sadržavati, koja je najveća adresa i koliko može biti najduža memorijska riječ.''

- *''ROM (Read Only Memory) je sklop s permanentno upisanim sadržajem. Ima samo jedan upis (najčešće u proizvodnji), a ostalo je čitanje. Ima više vrsta: ROM, Programmable ROM, Erasable PROM, Electrically EPROM.''*
- Postavili smo memorijski modul ROM na radnu površinu programa na isti način kako je objašnjeno u prethodnom zadatku.



- Memorijski modul ROM sadrži pinove:
  - **Pin A:** ulaz; određuje trenutno odabranu adresu koja se u sklopu obrađuje ili čita
  - **Pin D:** izlaz; na izlaz se šalje vrijednost zapisana na lokaciji koja se trenutno čita
  - **Pin SEL:** izlaz koji se koristi ako postoji više ROM modula(ako ih ima više u paraleli može omogućiti ili onemogućiti čitavi rad ROM modula ovisno o vrijednostima ''0'' ili ''1''), u suprotnom se može ignorirati.
- Broj memorijskih riječi možemo jednostavno povećati klikom na ROM modul. Zatim nam se na lijevoj strani prozora programa prikazuje tablica atributa u kojoj možemo postaviti željene vrijednosti (*Adress Bit Width, Data Bit Width*). Maksimalan broj riječi u ROM-u je 2^{24}=16.777.216, što je otprilike 16 MB podataka.
- Kako bi promijenili sadržaj ROM-a odlazimo u HEX Editor. *Navigacija:*desni klik na sklop→Edit Contents...



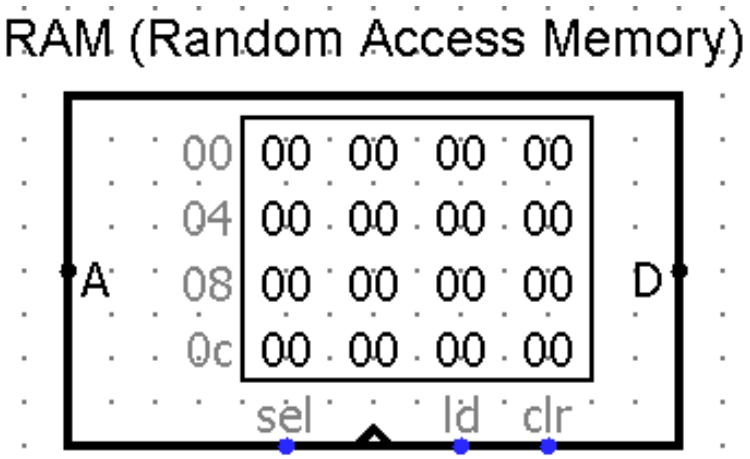
- Najveća moguća adresa je ffffff, a najduža memorijska riječ je 32 bita.

3) **''Postaviti na radnu površinu memorijski modul RAM i proanalizirati njegove ulaze i izlaze. Proučiti koliko memorijskih riječi može najviše sadržavati, koja je najveća adresa i koliko može biti najduža memorijska riječ.''**
    - *''RAM (Random Access Memory) je upisno/ispisna memorija koja pripada skupini memorija s izravnim pristupom. Sadržaj memorije RAM može se čitati i mijenjati-upisom novog sadržaja. Nedostatak: nestankom napajanja njezin se sadržaj briše.''*
    - Postavili smo RAM memorijski modul na radnu površinu programa te su njezini pinovi:
        - **Pin A:**ulaz; određuje trenutno odabranu adresu koja se u sklopu obrađuje ili čita
        - **Pin D:**ulaz/izlaz; ako je *pin Out* u stanju ''1'' ili neodređen, tada RAM šalje samo vrijednost trenutno odabrane adrese na *pin D*, koji je u tom slučaju izlaz. U slučaju da je *pin Out* u stanju ''0'' tada se *pin D* ponaša kao ulaz, kao vrijednost koja će biti postavljena na trenutno odabranu adresu sve dok se clock ne prebaci iz stanja ''0'' u ''1''.
        - **Pin Sel:**izlaz koji se koristi ako postoji više RAM memorijskih modula (omogućuje/onemogućuje u paraleli čitavi rad RAM modula ovisno o vrijednostima ''0'' ili ''1''), ako ne postoji može se ignorirati. Znači, kada je *pin Sel* u stanju ''0'', nikakva vrijednost se neće emitirati na *D izlazu* i vrijednost u memoriji se neće mijenjati sve dokse stanje na ulazu *Clock*ne prebaci iz ''0'' u ''1''.
        - **Clock**(ulaz za impulse): Kada je *pin Out* u stanju 0, a *Clock* ulaz se prebaci iz ''0'' u ''1''(istovremeno *pin Sel* je u ''1''/neodređeno i*pin Clr* u ''0''), tada se vrijednost trenutno odabrane adrese mijenja u stanje koje se nalazi na *D pinu*. Sve dok *Clock* ulaz ostaje ''0'' ili ''1'', vrijednost *D pina* neće biti pohranjena u RAM memoriji.
        - **Pin Out (ld):** ulaz; odabire koji bi RAM modul trebao emitirati vrijednost na *D pin*, ovisno o trenutno odabranoj adresi na *A pinu*. Te karakteristike izlaza su omogućene ako je *pin Out* u stanju ''1'' ili neodređen, a ako se nalazi u stanju

''0'', tada se *pin D* ponaša kao ulaz za učitavanje vrijednosti sve dok se vrijednost *Clock-a* ne prebaci iz ''0''u ''1''.

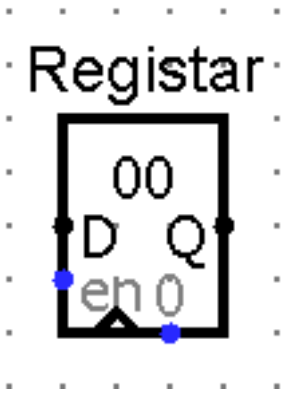
- **Pin Clr:** ulaz; kada je u stanju ''1'', a istovremeno je*pin Sel* u ''1'' ili neodređen, sve se vrijednosti u memoriji brišu, odnosno vrijednosti se postavljaju u ''0'', bez obzira na ostale impulse.

- Postavke RAM modula izvršavamo na isti način kao što je potrebno izvesti za ROM modul. Tako iz postavka u tablici atributa možemo vidjeti da je maksimalan broj riječu u RAM modulu 2^{24}=16.777.216 odnosno 16 MB podataka, najveća moguća adresa je ffffff, a najduža memorijska riječ je duljine 32 bita.

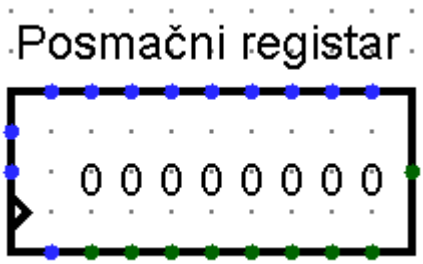


4) **''Postaviti na radnu površinu registar, proanalizirati njegove ulaze i izlaze. Koliko bita se može najviše pohraniti u registar.''**

- *''Registri pamte n-bitne podatke (n-broj bistabila). Ovisno o izvedbi mogu imati serijski upis/ispis podataka ili parelelni upis/ispis podataka. Za registre se koriste SR, JK i D bistabili.''*
- Postavili smo registar na radnu površinu programa:

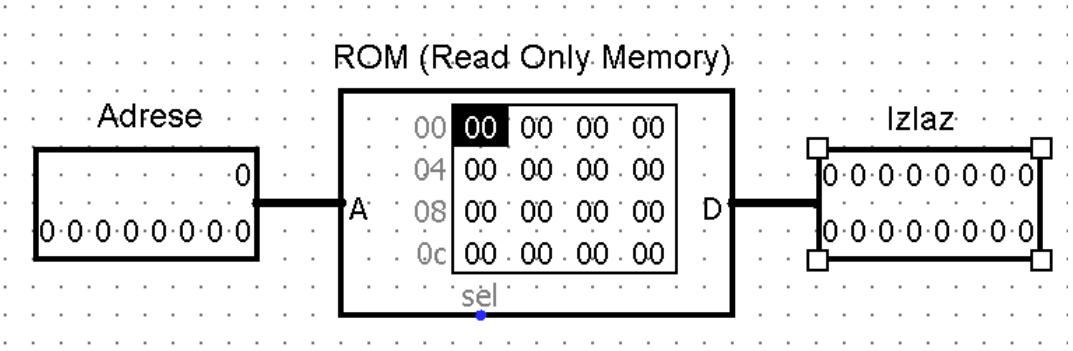


- Njegovi pinovi su:
    - **Pin D:** ulaz podataka; u trenutku kada se prebacivrijednost *Clock-a* iz stanja ''0'' u ''1'', vrijednost registra se prebacuje istog trenutka u vrijednost *ulaza D*.
    - **Pin Q:** izlaz; šalje vrijednosti trenutno pohranjene adrese u registar
    - **Clock**(ulaz za impulse): u trenutku kada se vrijednost ulaza prebaci iz ''0'' u ''1'', vrijednost registra će se obnoviti na vrijednost *ulaza D*.
    - **Pin Clr:** asinkroni reset; kada je u stanju ''0''/neodređeno, ulaz nema nikakav učinak na rad sklopa. Samo ako je u stanju ''1'' vrijednost registra se postavlja u ''0''. Taj asinkroni način rada *Clr-a* ne odnosi se na trenutnu vrijednost *Clock-a*. Sve dok je u stanju ''1'' ostali ulazi nemaju učinka.
- Maksimalan broj bitova koje možemo pohraniti u registar je 32-bita. U tablici atributa se još nalazi opcija za okidanje (na prednji brid, zadnji brid te na vrijednost napona - visoki/niski; u prijevodu ''0'' ili ''1'').
- *Dodatak:* u Logisim-u se nalazi još i posmačni (shift) registar.

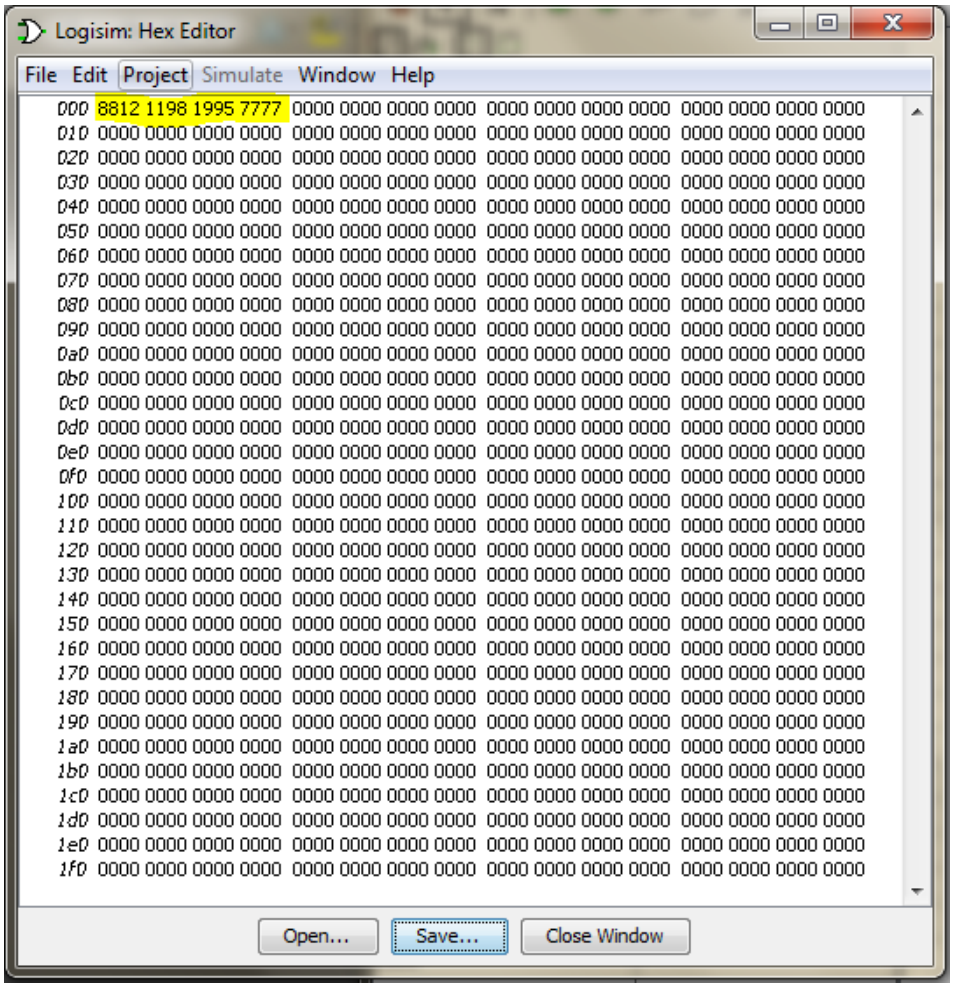


5) **''Projektirati memorijski modul ROM s 512 memorijskih riječi duljine 16 bita.Koliki je kapacitet tog memorijskog modula? Upišite sadržaj na prve četiri memorijske lokacije i simulirajte iščitavanje promjenom adrese.''**
    - Postavili smo ROM modul te namjestili postavke u tablici atributa. Postavili smo *Address Bit Width* na 9 bita (potrebno nam je 9-bitno adresiranje jer želimo postići 512 memorijskih riječi → 2^9=512), a *Data Bit Width* na 16 bita. Zatim smo postavili **Pin** na *ulaz A* te u njegovim postavkama postavili *Data Bits* u 9 bita. Isto to smo ponovili za *ulaz/izlaz D*, ali smo postavili *Data Bits* u 16 bita.

- *Realizacija sklopa:*



- Kapacitet modula izvedenog na ovaj način iznosi 1Kb.
- Zatim slijedi upisivanje podataka u ROM memoriju. Koristimo HEX Editor do kojeg dolazimo desnim klikom na ROM modul, zatim *Edit Contents...*
- Nakon što smo upisali sadržaj u ROM to izgleda ovako:

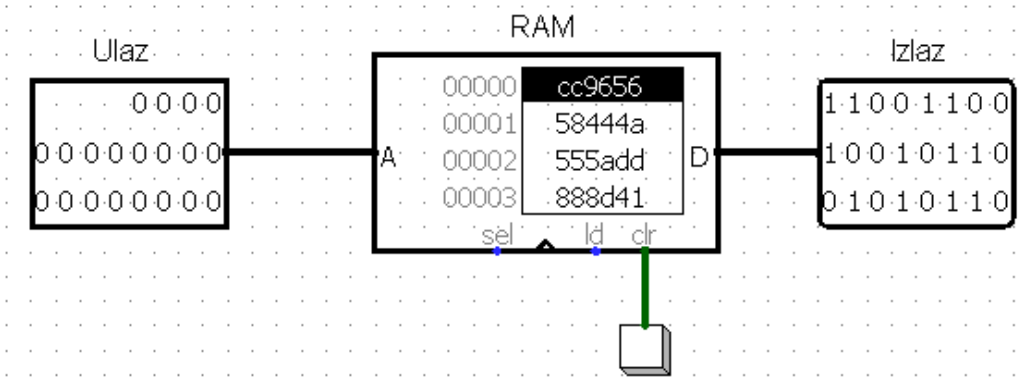


- Zatim krenemo iščitavati memorijske lokacije, mijenjajući adrese na *ulazu A*pomoću smo dobili rezultate:

Adresa (pin A-ulaz)	HEX	Binarno (pin D-izlaz)
000000000	8812	1000 1000 0001 0010
000000001	1198	0001 0001 1001 1000
000000010	1995	0001 1001 1001 0101
000000011	7777	0111 0111 0111 0111

6) **''Projektirati memorijski modul RAM s 1M memorijskih riječi duljine 24 bita. Koliki je kapacitet tog memorijskog modula? Upišite sadržaj za prve četiri memorijske lokacije i simulirajte iščitavanje promjenom adrese.''**

- Postavili smo RAM modul na radnu površinu te uredili njegovu konfiguraciju. Postavili smo dužinu riječi na 24 bita, a 1M smo realizirali preko 20 bita jer je 2^{20}=1.048.576 što je približno 1M.
- RAM memorijski modul projektiran na ovaj način ima kapacitet 25.165.824 bita (1.048.576 memorijskih riječi od kojih je svaka duljine 24 bita, pa je to ukupno 25.165.824 bita) odnosno otprilike 3 MB.
- *Realizacija sklopa:*



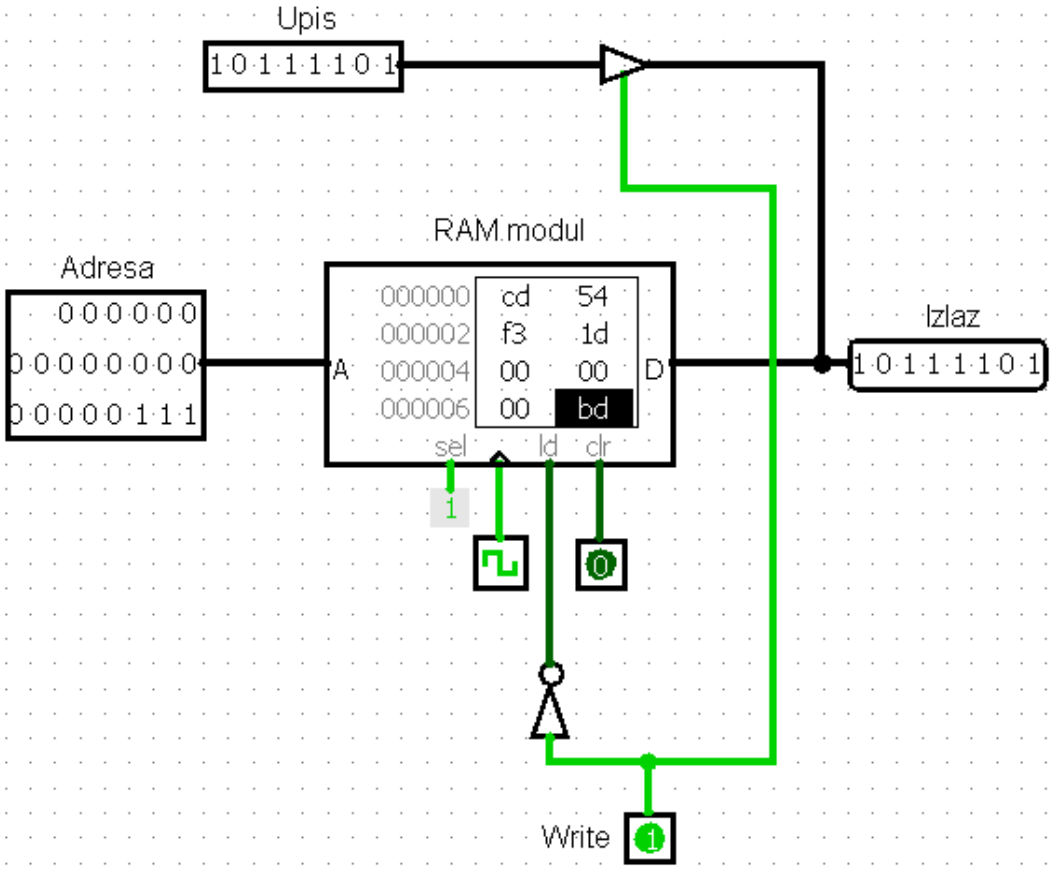
- *Rezultati:*

CLR	Adresa (pin A-ulaz)	HEX	Binarno (pin D-izlaz)
0	00000000000000000	cc9656	1100 1100 1001 0110 0101 0110
0	00000000000000001	58444a	0101 1000 0100 0100 0100 1010
0	00000000000000010	555add	0101 0101 0101 1010 1101 1101
0	00000000000000011	888d41	1000 1000 1000 1101 0100 0001
1	*(nije bitno)*	000000	0000 0000 0000 0000 0000 0000

- Iz tablice možemo vidjeti da postavljanjem *Clr-a* u ''1'' sav se sadržaj RAM memorije briše. Nešto slično kada se RAM briše nestankom napajanja (samo što je suprotno – iz ''1'' u ''0'').

7) **''Projektirati memorijski modul RAM s 4M memorijskih riječi duljine 8 bita. Koliki je kapacitet tog memorijskog modula? Simulirajte upisivanje podataka na prvih pet adresa promjenom adrese.''**

- Ponavljamo postupak iz prethodnog zadatka, samo što postavljamo duljinu memorijskih riječi na 8 bita, a 4M realiziramo s 22 bita jer je 2^{22}=4.194.304 (približno 4M).
- Sada je kapacitet RAM memorije 33.554.432 bita, odnosno 4.194.304 byte-a (33.554.432/8=4.194.304 jer 1 Byte ima 8 bitova).
- *Realizacija sklopa:*



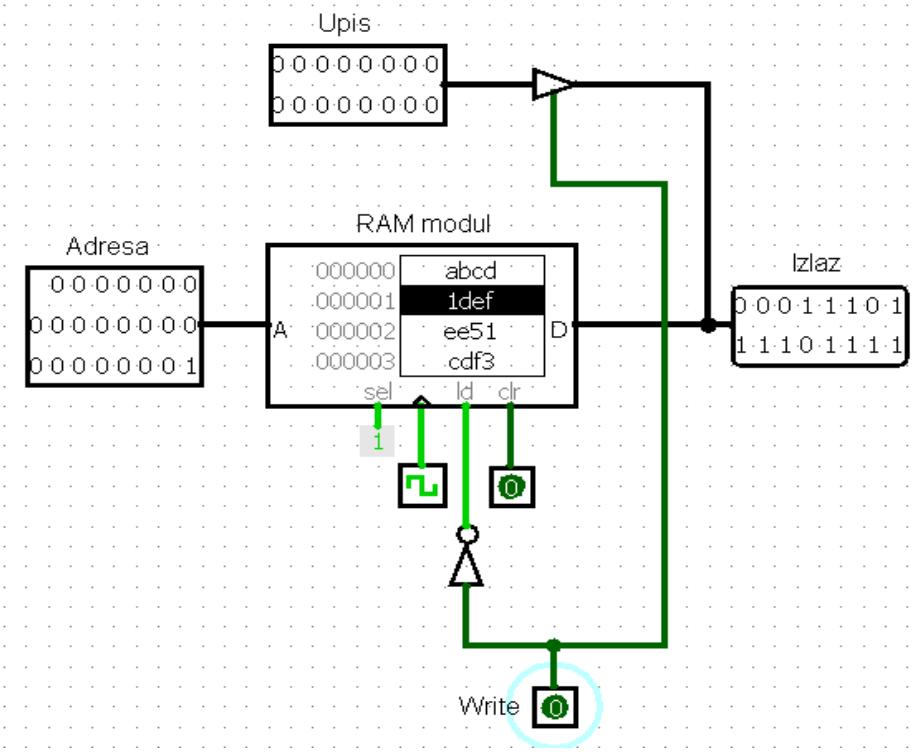
- Sklop radi kao i u prijašnjem zadatku, ali sada imamo i ''vanjsko'' upisivanje podataka. Vanjski upis podataka omogućujemo pomoću *pina Write* (ako je u ''1'' upis je omogućen, te se upisuje kada se vrijednost *Clock-a* dosegne ''1'', a kada je *Write* u ''0'' upis je onemogućen). Podatke upisujemo u *pin Upis*, a mijenjanjem *Adrese* odabiremo na koju memorijsku lokaciju želimo upisati novi sadržaj. Na *pinu Izlaz* dobivamo binarnu kombinaciju sadržaja na odabranoj adresi. Istovremeno *pin Sel* mora biti u ''1'', a *Clr* u ''0''.

- *Rezultati:*

Write	*Upis*	Adresa	HEX	Izlaz
0	(*nije bitno*)	00000000000000000000	cd	1100 1101
0	(*nije bitno*)	00000000000000000001	54	0101 0100
0	(*nije bitno*)	00000000000000000010	f3	1111 0011
0	(*nije bitno*)	00000000000000000011	1d	0001 1101
1	1011 1101	00000000000000000111	bd	1011 1101

8) "Projektirati memorijski modul RAM sa 8M memorijskih riječi duljine 16 bita. Koliki je kapacitet tog memorijskog modula? Simulirajte sustav koji omogućuje i upisivanje i iščitavanje podataka na prvih pet adresa."

- Postavimo RAM modul te njegove postavke u tablici atributa. Duljinu riječi postavimo na 16 bita, a 8M memorijskih riječi realiziramo pomoću 23 bita (2^{23}=8.388.608; približno 8M). Tako postavimo i sve ostale komponente na radnoj površini programa (osim adrese).
- Nakon što smo sve postavili, sada RAM modul ima kapacitet 134.217.728 bita što je jednako 16.777.216 bytea; približno 16 MB.
- Zadatak je zapravo vrlo sličan, tj. shema spoja je ista, samo što ćemo sada upisati prvih 5 lokacija, a prije sam radi primjera upisivao 4 memorijske lokacije u RAM modulu.
- *Realizacija sklopa:*



- *Rezultati:*

Write	*Upis*	Adresa	HEX	Izlaz
1	1010 1011 1100 1101	00000000000000000000000	abcd	1010 1011 1100 1101
1	0001 1101 1110 1111	00000000000000000000001	1def	0001 1101 1110 1111
1	1110 1110 0101 0001	00000000000000000000010	ee51	1110 1110 0101 0001
1	1100 1101 1111 0011	00000000000000000000011	cdf3	1100 1101 1111 0011
1	0001 1001 1001 1101	00000000000000000000100	199d	0001 1001 1001 1101

9) ''Projektirati memorijski sustav od četiri memorijska modula RAM ukupnog kapaciteta 64MB. Svi moduli su jednakog kapaciteta i duljine memorijske riječi 8 bita.''

- Postavili smo 4 RAM modula ukupnog kapaciteta 64 MB (svaki modul posebno mora imati 16 MB). To smo postigli tako što smo na svakom modulu postavili 24 bitno adresiranje (2^{24}=16.777.216). Isto tako na svakom modulu smo postavili 8 bita duljinu memorijskih riječi.
- Postupak spajanja: svaki modul ima zajednički ulaz (*pin A*), zajednički *Clock*, izlaz (*pin D*) i *pin ld(Out)*. Također svaki modul mora imati *pin Sel* u ''1''. Kako je shema prilično slična shemi iz prethodnog zadatka (osim što je potrebno 4 RAM modula), tako je i sam rad sklopa identičan, tj. upis i ispis podataka.
- U shemu sam još dodao LED matrix sklop koji omogućuje da pomoću njega ''grafički'' vidimo binarni zapis na izlazu.
- Ukoliko želimo gledati sadržaj RAM memorije (bez korištenja HEX editora) preko sklopova, moramo postaviti *pin Write* u ''0''. Ova činjenica se također odnosi na prethodne zadatke. *Clock* je također bitan jer bez njega ne možemo upisati sadržaj izvana (potreban je jedan impuls za upis novog sadržaja).
- Shema se zbog svoje veličine nalazi na idućoj stranici.

- *Realizacija sklopa:*