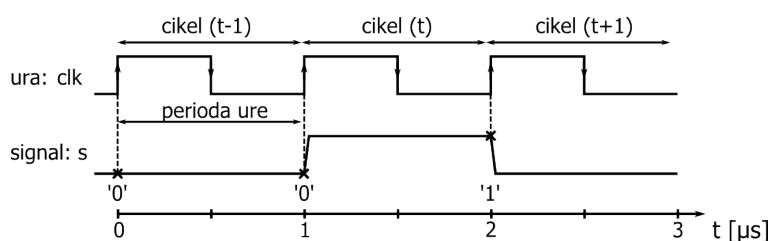


5

Sekvenčni gradniki in vezja

Sekvenčna vezja so logična vezja, pri katerih je izhod odvisen od vhodov in stanja shranjenega v pomnilnih elementih. Osnovni gradniki shranijo le en bit (logično 0 ali 1), sestavljeni pa shranijo večbitno vrednost. *Stanje vezja* predstavljajo vse logične vrednosti, ki so v nekem trenutku shranjene v vezju.

V sekvenčnih vezjih imamo poseben signal, ki določa, kdaj naj se izhodi spremenijo. Takšen signal je v obliki periodičnih impulzov in ga označujemo z imenom ura (angl. clock, clk). Uro dobimo iz posebnega vezja, ki se imenuje oscilator. V digitalnih sistemih bomo najpogosteje naleteli na kvarčne oscilatorje, s katerimi dobimo stabilno uro natančno določene frekvence.



Slika 5.1: Proženje na nivo ali na fronto ure.

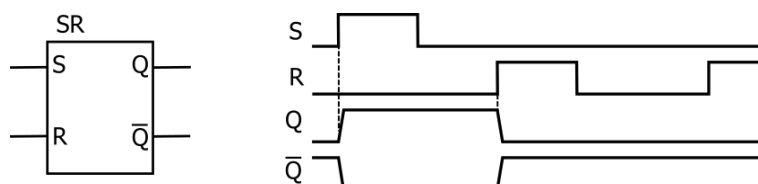
Slika 5.1 prikazuje primer časovnega diagrama urnega signala. Glavni parameter ure je perioda, ki predstavlja časovni interval med sosednjimi impulzi. Časovni interval ene periode, v katerem se stanje spremeni iz nizkega v visoko (ali obratno), imenujemo *urni cikel*. Včasih je pomembno tudi razmerje med trajanjem visokega in nizkega dela cikla, ki ga podajamo v odstotkih. Obratna vrednost periode je frekvenca, v primeru s slike velja: $f = 1/1 \mu s = 1 \text{ MHz}$. Trenutek prehoda iz nizkega v visoko stanje imenujemo prednja ali naraščajoča fronta ure. Prehod iz visokega v nizko stanje pa imenujemo zadnja ali padajoča fronta. Fronte ure lahko na časovnem diagramu posebej označimo s puščicami.

5.1 Zapah in flip-flop



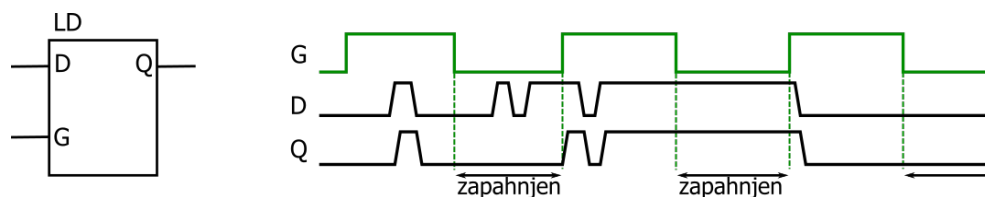
Zapah (angl. latch) je pomnilni gradnik, ki shrani nastavljeno logično stanje. Najpreprostejši je zapah SR z dvema kontrolnima vhodoma: S (angl. Set) in R (angl. Reset) ter izhodom Q in negiranim izhodom \bar{Q} . Kadar sta vhoda na 0, izhoda ohranjata stanje:

- $S=1$ in $R=0$ povzroči, da gre Q na 1 (\bar{Q} na 0);
- $S=0$ in $R=0$ ohranja stanje izhoda;
- $S=0$ in $R=1$ povzroči, da gre izhod Q na 0 (\bar{Q} na 1);
- $S=0$ in $R=0$ ohranja stanje izhoda.



Slika 5.2: Zapah SR in časovni potek izhodnih signalov.

Zapah SR lahko spravimo v nestabilno stanje, če hkrati postavimo oba vhoda na 1. Osnovni stabilen gradnik je zapah D, ki ima podatkovni vhod D , kontrolni vhod G (angl. Gate, vrata) in izhod Q . Kadar je $G = 1$, se stanje s podatkovnega vhoda prenese na izhod. Pravimo, da je zapah transparenten in vse spremembe vhoda z majhno zakasnitvijo prenaša na izhod. Ko gre signal G na 0, se stanje izhoda zapahne in ohranja zadnjo vrednost.



Slika 5.3: Zapah D: simbol in časovni diagram.

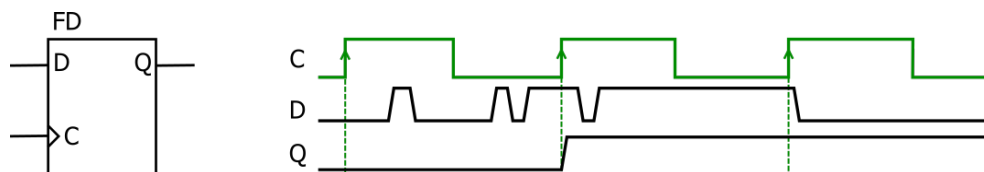
Če je časovni potek krmilnega signala G periodičen, kot na sliki 5.3, bo izhod zapaha polovico periode konstanten oz. zapahnen. Med visokim delom periode ure se na izhod prenašajo vse spremembe iz podatkovnega vhoda, tudi kratka nihanja signala, ki predstavljajo motnje. Nihanja signala med periodo ure odstranijo pomnilni elementi z imenom *flip-flopi*, ki spreminjajo izhod le ob fronti ure.

Podatkovni flip-flop (angl. Data Flip-Flop, oznaka FD ali DFF) prenaša logično stanje z vhoda D na izhod Q ob fronti ure C (ali CLK , ang. clock). Flip-flop pri katerem se izhod spreminja ob naraščajoči fronti ure (angl. rising edge), opišemo s pogojnimi stavkom:

```

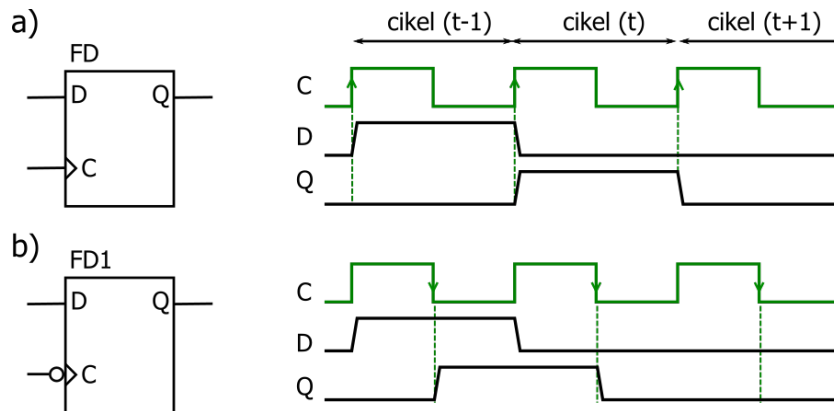
if rising_edge(clk) then
  q <= d;
end if;

```



Slika 5.4: Simbol in časovni diagram podatkovnega flip-flopa.

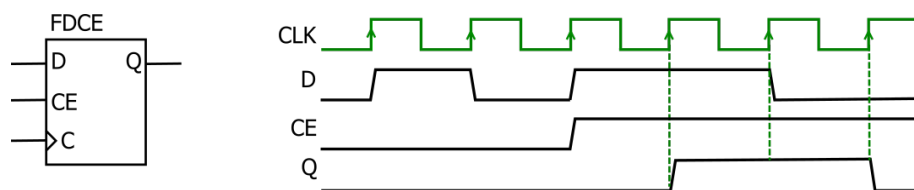
Skupna lastnost flip-flopov je, da se izhodni signal spreminja le ob prednji ali zadnji fronti ure, kot prikazuje slika 5.5. Takšne pomnilne gradnike imenujemo *sinhroni* gradniki in imajo signal na katerega je vezana ura v simbolu označen s trikotnikom, če so proženi na zadnjo fronto ure pa je pred njim še krožec (negacija ure). Analiza vezja s sinhronimi gradniki je preprostejša kot analiza vezja z zapahi, zato pri snovanju vezij pogosteje uporabljamo flip-flope.



Slika 5.5: Potek signalov flip-flopa D a) na naraščajočo in b) padajočo fronto ure

Signali v sinhronem vezju se spreminjajo le ob aktivni fronti ure, npr. tik za naraščajočo fronto ure, kot prikazuje časovni diagram na sliki 5.5a. Flip-flop prenese na izhod vrednost vhoda iz prejšnjega urnega cikla: $Q(t) = D(t - 1)$, $Q(t + 1) = D(t)$ itn.

Podatkovni flip-flop v sinhronih vezjih shrani stanje za en cikel ure. Če želimo stanje izhoda zadržati dlje, uporabimo flip-flop z dodatnim vhodom za omogočanje ure z oznako CE (angl. Clock Enable). Sinhroni flip-flop ima lahko tudi asinhrono, od ure neodvisne, kontrolne vhode, npr. reset (rst) ali clear (clr), ki postavi izhod takoj v stanje 0.

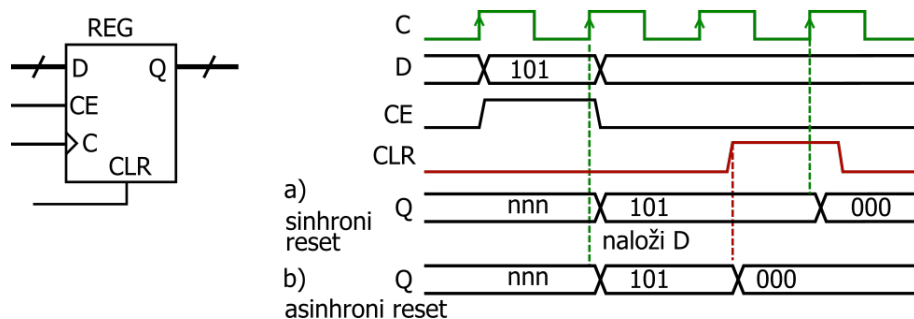


Slika 5.6: Simbol in časovni diagram flip-flopa, ki ohranja stanje izhoda.

5.2 Register



Pomnilni gradnik, ki shranjuje večbitne besede, se imenuje register. Sestavljen je iz flip-flopov s signalom za omogočanje ure CE . Kadar je CE na 0, bo izhod registra ohranjal vrednost: $Q(t+1) = Q(t)$, ko je CE na 1 pa se vrednost prenese na izhod: $Q(t+1) = D(t)$.



Slika 5.7: Simbol in časovni diagram registra s signalom reset.

Register ima lahko tudi kontrolni signal (CLR), ki resetira oz. postavi na 0 vse bite izhoda. Glede na delovanje tega signala ločimo registre z asinhronim in registre s sinhronim reset signalom. Sinhroni reset signal vpliva na izhod ob fronti ure (slika 5.7a), asinhroni reset pa takoj, ko ga postavimo na 1, ne glede na fronto ure (slika 5.7b). Signal reset ima prednost pred ostalimi kontrolnimi signali, zato ostane izhod na 0, dokler je reset aktiven.

Opis registra z asinhronim reset signalom:

```

if clr = '1' then
  q <= "0000";
elsif rising_edge(clk) and ce='1' then
  q <= d;
end if;

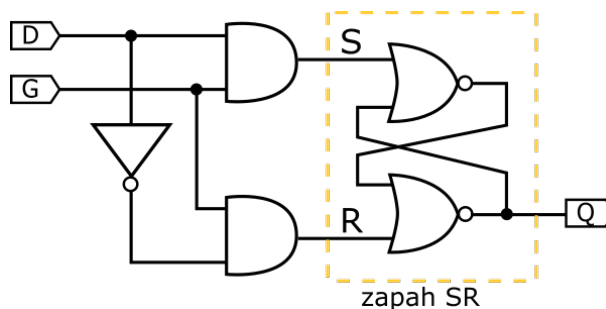
```

5.3 Izvedba pomnilnih gradnikov



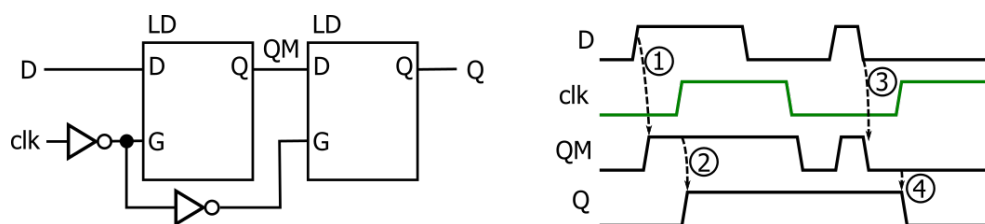
Pomnilne gradnike lahko naredimo z vezjem iz osnovnih logičnih vrat, če pripeljemo izhod vezja nazaj na vhod, kar imenujemo *povratna vezava*. Povratna vezava omogoča ohranjanje stanja izhoda.

Z vezjem iz dveh logičnih vrat NOR (OR z negatorjem na izhodu) naredimo zapah SR, z nekaj dodatnimi vrati pa podatkovni zapah, kot prikazuje slika 5.8.



Slika 5.8: Izvedba zapaha D z logičnimi vrati.

Flip-flop naredimo z zaporedno vezavo dveh zapahov in invertiranjem kontrolnega vhoda. Slika 5.9 prikazuje vezavo in časovni diagram, na katerem bomo razložili delovanje flip-flopa. Prvi zapah je transparenten, ko je signal ure (clk) na 0, ob prehodu ure na 1 pa zapahne zadnjo vrednost (1). Obenem postane transparenten drugi zapah, ki prenese vrednost na izhod (2). Če pride do spremembe na podatkovnem vhodu, ko je ura na 0, se sprememba sicer prenese na izhod prvega zapaha (3), ne pa tudi na izhod vezja. Do spremembe na izhodu vezja pride šele ob naslednji naraščajoči fronti ure (4), tako da dobi izhod vrednost s podatkovnega vhoda tik pred fronto ure.



Slika 5.9: Izvedba flip-flopa D iz dveh zapahov.

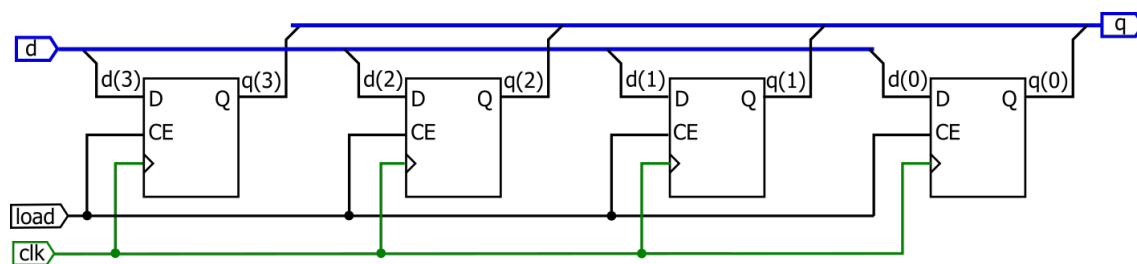
Zapah iz petih logičnih vrat in flip-flop iz dveh zapahov sta zelo potratna elementa za shranjevanje enega bita, zato so v digitalnih integriranih vezjih pomnilni gradniki narejeni z bolj optimalnimi vezavami na nivoju tranzistorjev.

5.4 Vezja s flip-flopi



Štiri-bitni register

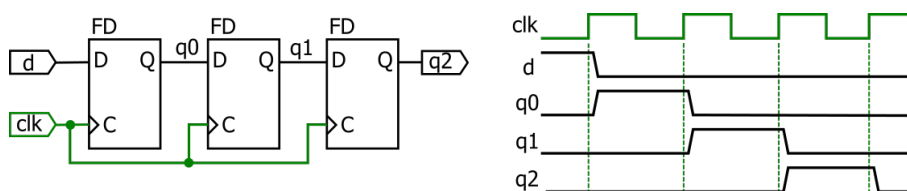
Štiri-bitni register naredimo iz štirih flop-flopov FDCE s skupaj vezanimi kontrolnimi vhodi CE in uro, kot prikazuje slika 5.10. Vsak flip-flop bo ob fronti ure in aktivnem signalu $load$ naložil en bit iz podatkovnega vhoda na izhod. Ker posamezni flip-flopi prenašajo podatek na izhod vzporedno, pravimo tej vezavi *vzporedna vezava*.



Slika 5.10: Vezje 4-bitnega registra z vzporedno vezavo flip-flopov.

Pomikalni register

Z *zaporedno vezavo* povežemo podatkovne flip-flope drugega za drugim, tako da je podatkovni izhod predhodnjega povezan z vhom naslednjega. Tudi pri tej vezavi so kontrolni signali (npr. ura) vezani skupaj. Slika 5.11 prikazuje zaporedno vezavo treh flip-flopov in tipičen časovni diagram na katerem vidimo, kako se signal iz vhoda zakasni na poti proti izhodu.



Slika 5.11: Zakasnitev signalov pri prehodu čez zaporedne flip-flope D.

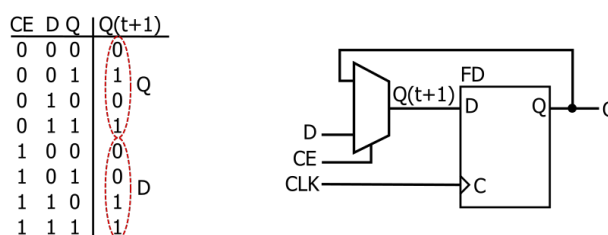
Vezje imenujemo pomikalni register, ker se sprememba vrednosti na vhodu postopoma pomika proti izhodu. Izhodni signal 3-bitnega pomikalnega registra je zakasnen za tri urne cikle: $Q(t + 3) = D(t)$.

Enobitni register

Z osnovnim flip-flopom D in logiko bomo naredili enobitni register. Delovanje registra opišemo z enačbo:

$$Q(t+1) = \begin{cases} D(t) & CE = 1 \\ Q(t) & \text{sicer} \end{cases}$$

Enačbo predstavimo s pravilnostno tabelo, v kateri so na levi strani vse kombinacije vhodov CE , D in stanja Q , na desni strani pa je signal $Q(t+1)$, ki določa naslednje stanje. Tabela je identična pravilnostni tabeli izbiralnika 2-1, ki predstavlja logiko na vhodu flip-flopa.



Slika 5.12: Tabela in zgradba enobitnega registra.

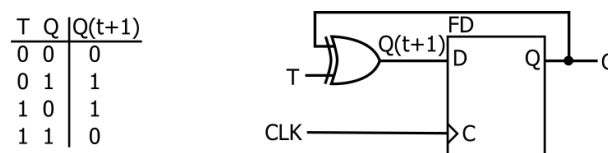
Zadrževanje vrednosti bi dosegli tudi z blokiranjem ure flip-flopa z logiko ali elektronskim stikalom pred urnim vhodom. Blokiranje ure (angl. clock gating) zahteva natančen časovni potek kontrolnega vhoda, da ne prihaja do neželenih proženj flip-flopa. Uporablja se predvsem v namenskih integriranih vezjih.

Preklopni flip-flop

Preklopni flip-flop ali flip-flop T (angl. Toggle) ima krmilni signal T . Kadar je $T = 1$, naj izhod ob naraščajoči fronti ure zamenja vrednost, sicer pa se vrednost ohranja:

$$Q(t+1) = \begin{cases} \text{NOT } Q(t) & T = 1 \\ Q(t) & \text{sicer} \end{cases}$$

Preklopni flip-flop naredimo iz podatkovnega flip-flopa in logike na vhodu. Logika določa stanje naslednjega izhoda $Q(t+1)$ v odvisnosti od trenutnega vhoda T in stanja $Q(t)$.

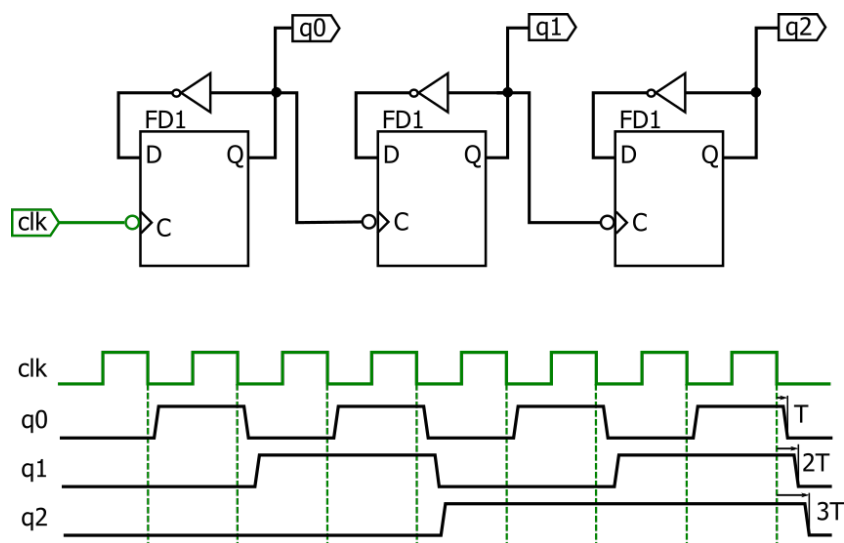


Slika 5.13: Pravilnostna tabela in zgradba flip-flopa T.

Pravilnostna tabela določa logično funkcijo na vhodu flip-flopa, ki je v našem primeru XOR. Če potrebujemo flip-flop, ki stalno preklaplja izhodno stanje, se logika poenostavi v inverter med izhodom in vhodom.

Serijski števec

Iz zaporedno vezanih preklonnih flip-flopov naredimo serijski števec, kot prikazuje slika 5.14. Gradniki so vezani tako, da izhod posameznega flip-flopa krmili uro naslednjega. Na časovnem diagramu vidimo, da predstavlja zaporedje izhodnih signalov q_2 , q_1 in q_0 binarno zaporedje, ki se spreminja ob ciklih vhodne ure. V realnem vezju je izhod prvega flip-flopa q_0 za čas T zakasnen za fronto ure. Ta signal krmili naslednji flip-flop, katerega izhod ima že dvojno zakasnitev ($2T$) glede na vhodno uro, izhod tretjega pa trojno zakasnitev ($3T$).



Slika 5.14: Časovni potek signalov 3-bitnega serijskega števca.

Števec pogosto uporabljamo kot generator zaporedja binarnih vrednosti. Če opazujemo vse izhode hkrati, nam različne zakasnitve preklonov povzročajo težave pri dekodiranju vrednosti in omejujejo navišjo frekvenco delovanja števca. Zaradi tega števce raje načrtujemo v obliki sinhronnega vezja, kjer so vsi flip-flopi vezani na isto uro.

5.5 Pomnilnik

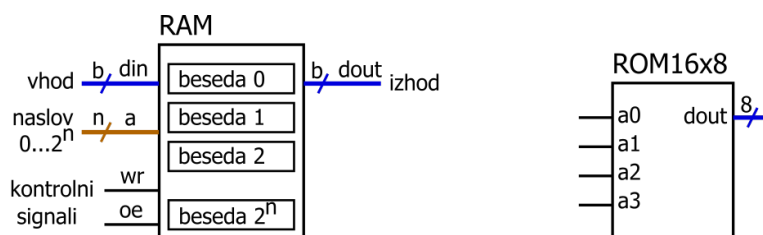


Pomnilniki so elementi za shranjevanje podatkov in ukazov, ki jih npr. obdelujejo računalniški digitalni sistemi. Spoznali smo že registre v katerih hranimo večbitne besede, pomnilniki pa vsebujejo pomnilne celice za množico večbitnih besed. Pomnilnik z $m \times n$ pomnilnimi celicami vsebuje m n -bitnih besed, npr. pomnilnik z matriko 1024×8 vsebuje 1024 oz. $1k$ besed velikosti 8-bitov (1 bajt). Produkt $m \times n$ imenujemo *kapaciteta pomnilnika*. Podatke prenašamo v ali iz pomnilnika po besedah. Pri večini pomnilnikov lahko dostopamo le do ene besede naenkrat.

Elektronske polprevodniške pomnilnike delimo na bralne pomnilnike, bralno-pisalne in trajne (angl. non-volatile) pomnilnike. Bralni pomnilniki s kratico ROM (angl. Read Only Memory) imajo vnaprej zapisano vsebino in iz njih lahko le beremo podatkovne besede. Namenjeni so za shranjevanje podatkov, ki jih v tovarni zapišemo in jih v vezju ne bomo spreminjali. Bralno

pisalni pomnilniki omogočajo branje in zapisovanje podatkov. Pomnilniki z oznako RAM (angl. Random Access Memory) hranijo zapisano vsebino dokler imajo napajanje. Oznaka pomeni, da imajo enak čas za dostop do naključno izbrane pomnilniške besede, za razliko od prvih pomnilnikov z zapisom na magnetnem traku, kjer je bil čas dostopa do besede odvisen od trenutnega položaja bralne glave.

Trajni pomnilnik vsebuje bralno-pisalne pomnilne celice (EPROM, angl. Erasable Programmable ROM), ki omogočajo ohranjanje podatkov brez prisotnosti napajanja. Primer trajnega pomnilnika je tudi bliskovni pomnilnik z oznako FLASH.



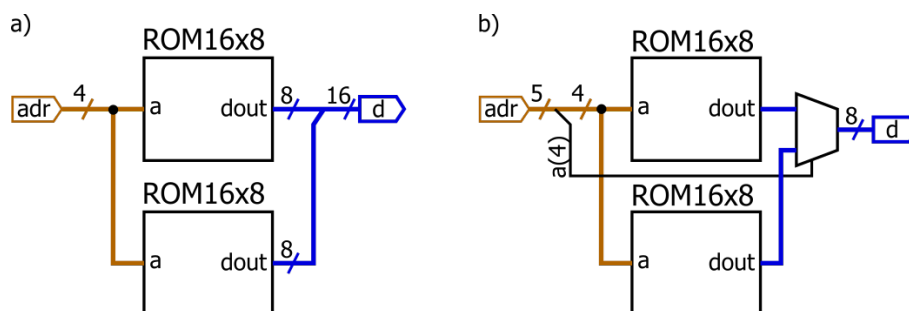
Slika 5.15: Blokovna shema pomnilnika RAM in ROM

Signale na priključkih pomnilnika delimo na:

- podatkovno vodilo (ang. *data*), ki je lahko ločeno na podatkovni vhod (*din*) in podatkovni izhod (*dout*),
- naslovno vodilo (ang. *address*) s katerim izberemo posamezno besedo in
- krmilne signale, ki določajo operacije (branje, pisanje, mirovanje).

Pomnilnik, ki ima n -bitno naslovno vodilo, vsebuje 2^n pomnilnih besed. Posamezno besedo izberemo tako, da postavimo na naslovno vodilo binarno število med 0 in $2^n - 1$. S krmilnimi signali pa določimo, ali bomo v pomnilnik naložili novo besedo ali pa bomo brali iz pomnilnika.

Pomnilnik z večjo kapaciteto dobimo tako, da skupaj vežemo več manjših pomnilnikov. Shema na sliki 5.16a prikazuje izdelavo pomnilnika ROM16x16, shema 5.16b pa pomnilnika ROM32x8 iz dveh manjših pomnilnikov ROM16x8.



Slika 5.16: Blokovna shema pomnilnika ROM

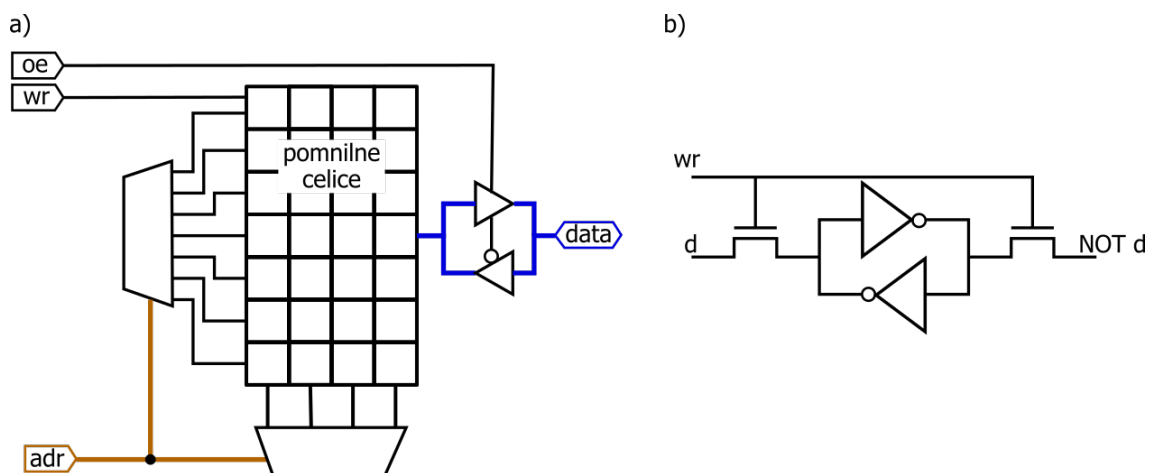
Pomnilnik ROM lahko uporabimo kot splošen gradnik za izdelavo kombinacijskih vezij, podobno kot izbiralnik. Vsebina pomnilnika mora biti enaka desni strani pravilnostne tabele kom-

binacijskega vezja, vhodi vezja pa so vezani na naslovne signale pomnilnika. Takšne bralne pomnilnike zasledimo v programirljivih vezjih. Imenujemo jih tudi vpogledna tabela (LUT, angl. Look-Up Table) in imajo tipično 3 do 6 naslovnih vhodov in en izhod.

5.6 Izvedba statičnega pomnilnika



Slika 5.17 prikazuje zgradbo statičnega pomnilnika RAM in posamezne pomnilne celice.

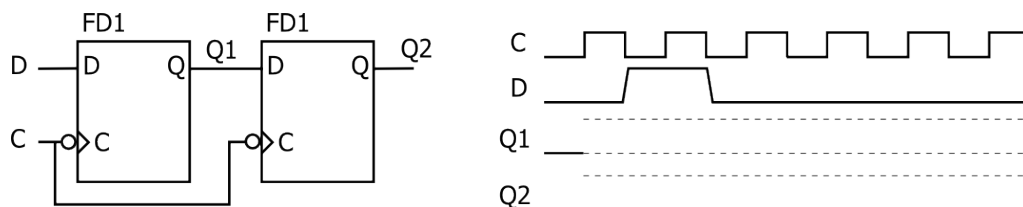


Slika 5.17: Izvedba statičnega pomnilnika in pomnilna celica.

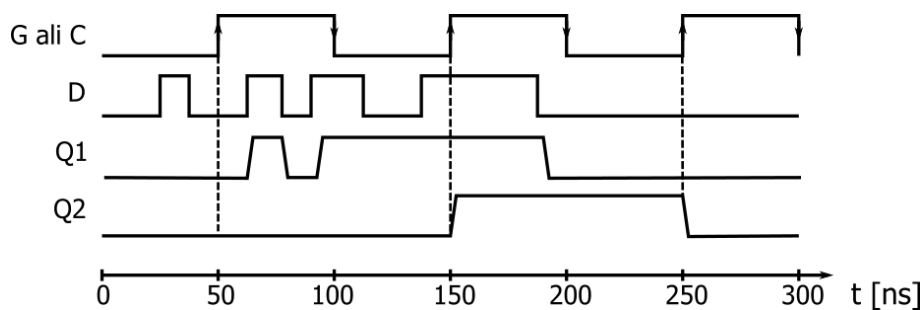
Razdeljevalniki poskrbijo za izbiro pomnilnih celic, ki so razporejene v matriko. Vsaka celica je sestavljena iz dveh negatorjev za shranjevanje podatka in transistorjev za vpis nove vrednosti. Za vsak bit potrebujemo skupaj 6 transistorjev, kar je precej manj kot pri izvedbi s flip-flopom D. Še bolj učinkoviti so dinamični pomnilniki, ki hranijo zapis v enem samem transistorju, vendar imajo zahtevnejšo krmilno logiko.

Naloge

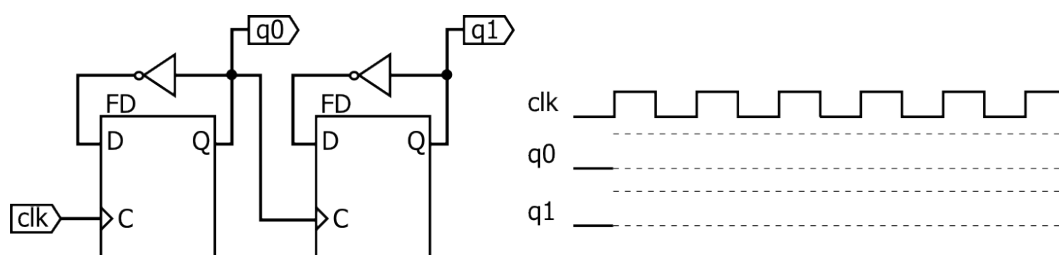
1. Vriši v časovni diagram spreminjanje izhodnih vrednosti dveh zaporednih flip-flopov, ki sta prožena na zadnjo fronto ure.



2. Ugotovi kateremu elementu (flip-flop, zapah) pripadata izhodna signala $Q1$ in $Q2$ na časovnem diagramu.



3. Določi časovni potek izhodnih signalov 2-bitnega števca, ki je sestavljen iz flip-flopov na prednjo fronto ure in negatorjev.



4. Ugotovi, koliko vpoglednih tabel in kakšne bi potrebovali za izdelavo 5-bitnega sinhronega števca.