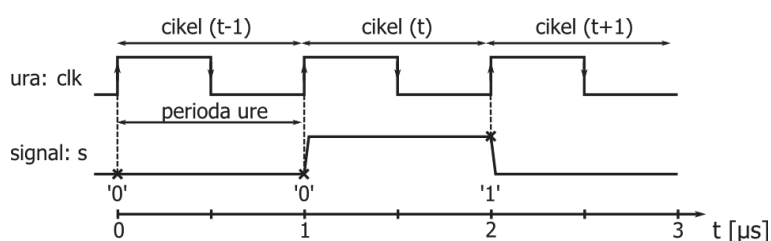


5

Sekvenčna vezja

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

V sekvenčnih vezjih imamo običajno 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.

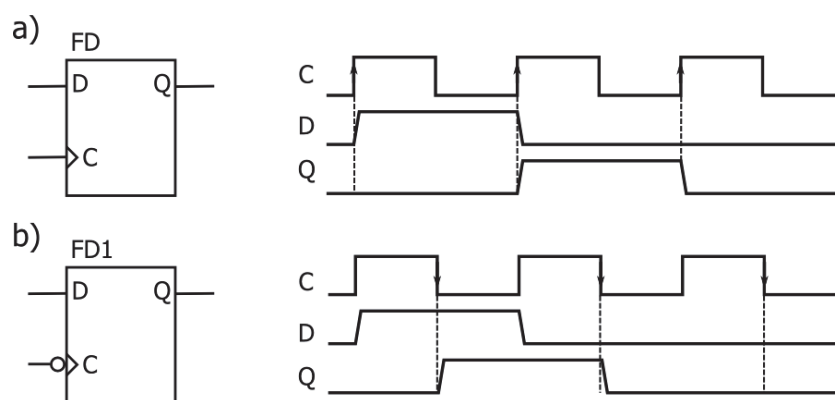
Slika 5.1 prikazuje primer časovnega diagrama urnega signala. Glavni parameter ure je perioda, ki predstavlja časovni interval med sosednjimi impulzi. Časovni interval ene periode, v katerem se stanje spremeni iz nizkega v visoko (ali obratno), imenujemo *urni cikel*. Včasih je pomembno tudi razmerje med trajanjem visokega in nizkega 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 Flip-flop



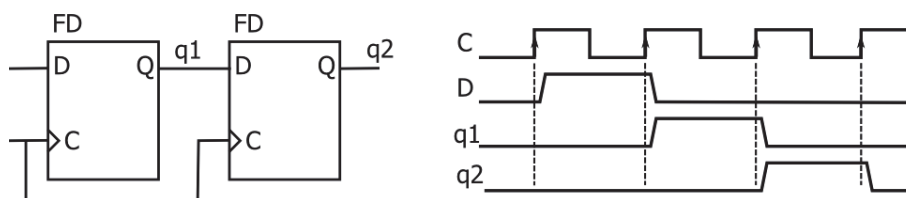
Pomnilni elementi, ki spreminjajo stanje ob taktu ure, se imenujejo *flip-flopi*. Podatkovni flip-flop (angl. Data Flip-Flop, DFF) prenaša logično stanje z vhoda D na izhod Q ob fronti ure. Vhodni podatkovni signal določa vrednost, ki bo na izhodu vezja v naslednjem urnem ciklu: $Q(t) = D(t - 1)$, $Q(t + 1) = D(t)$ itn.



Slika 5.2: Simbola in časovna diagrama podatkovnih flip-flopov.

Skupna lastnost flip-flopov je, da se izhodni signal spreminja le ob prednji ali zadnji fronti ure. Podatkovni flip-flop, ki je prožen na zadnjo fronto ure, je označen z negacijo (krožcem) pred vhodom za uro.

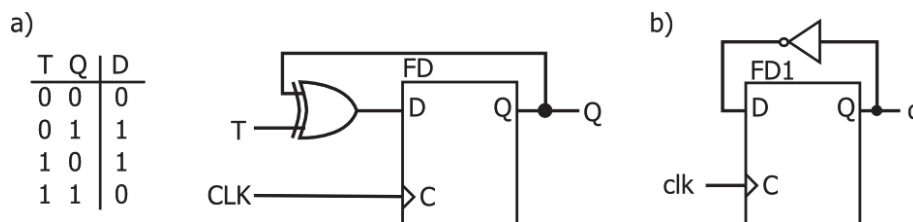
Če vezemo drug za drugim dva podatkovna flip-flopa, bomo zakasnili signal za dva cikla, kot prikazuje slika 5.3. Takšna vezava je osnova za izvedbo elementov, ki jih imenujemo pomikalni registri.



Slika 5.3: Zakasnitev signalov pri prehodu čez zaporedne flip-flope D.

Preklopni flip-flop ali flip-flop T (angl. Toggle) ima krmilni signal T , ki povzroči, da izhod ob naraščajoči fronti ure zamenja vrednost: $Q(t + 1) = \text{NOT } Q(t)$. Kadar je signal T na 0, izhod ohranja vrednost. Preklopni flip-flop naredimo iz podatkovnega flip-flopa in logike na

vhodu. Podatkovni flip-flop izvaja funkcijo: $Q(t + 1) = D(t)$, logika na vhodu pa določa $D(t)$ v odvisnosti od vhoda T in stanja $Q(t)$, kar opišemo s pravilnostno tabelo.

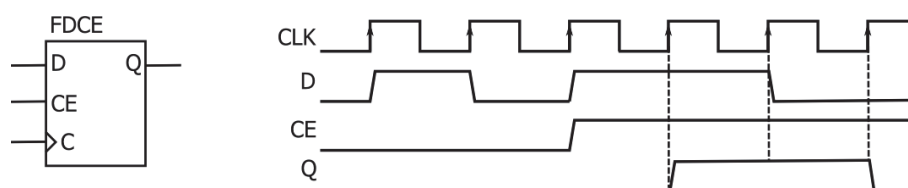


Slika 5.4: a) tabela in zgradba flip-flopa T, b) flip-flop s stalnim preklapljanjem izhoda.

Pravilnostna tabela določa logično funkcijo na vhodu flip-flopa, ki je v našem primeru XOR. Če potrebujemo flip-flop, ki stalno preklaplja izhod, uporabimo inverter med izhodom in vhodom flip-flopa (slika 5.4b).

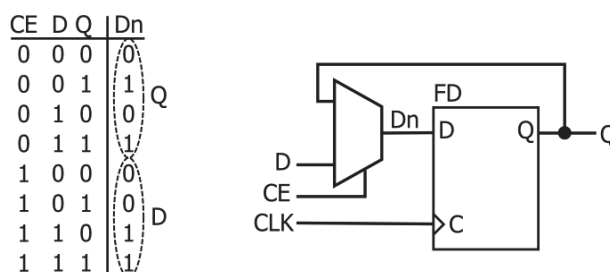
5.2 Register

Register je podatkovni pomnilni gradnik s signalom za omogočanje CE . Kadar je CE na 0, bo izhod registra ohranjal vrednost.



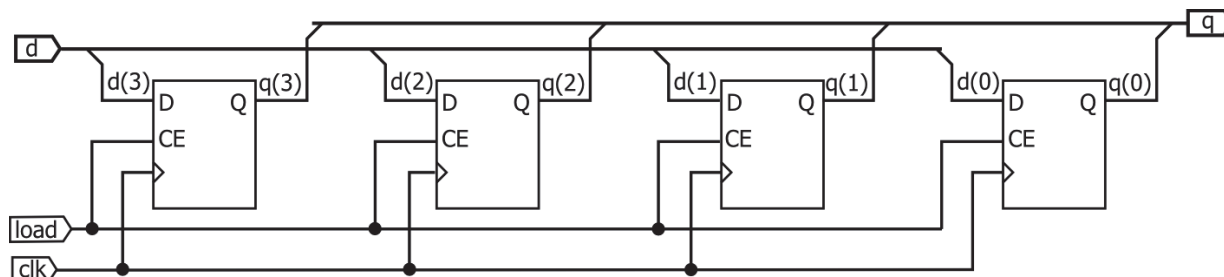
Slika 5.5: Simbol in časovni diagram flip-flopa, ki ohranja stanje izhoda.

Ohranjanje vrednosti izhoda opisuje enačba: $Q(t + 1) = Q(t)$. Delovanje registra opišemo s tabelo, v kateri so na levi strani vse kombinacije vhodov CE , D in stanja Q , na desni strani pa je signal Dn , ki določa naslednje stanje $Q(t + 1)$. Tabela je identična pravilnostni tabeli izbiralnika 2-1, ki predstavlja logiko na vhodu flip-flopa.



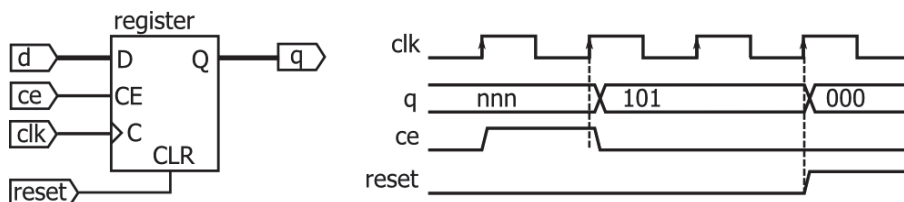
Slika 5.6: Tabela in zgradba enobitnega registra.

Zadrževanje vrednosti bi dosegli tudi z elektronskim stikalom, ki bi odklopilo uro. Takšna rešitev za sinhrona vezja ni najboljša, zato se na vhodu registra pogosteje uporablja izbiralnik. Register, ki shranjuje večbitne besede, naredimo z vzporedno vezavo flip-flopov FDCE. Slika 5.7 prikazuje logično shemo 4-bitnega registra z uro in signalom *Load* za nalaganje nove vrednosti.



Slika 5.7: Shema 4-bitnega registra.

Flip-flopi v registru imajo pogosto tudi kontrolni signal *CLR*, ki deluje kot reset; ko ga postavimo na 1, gre izhod na 0, ne glede na fronto ure. Pravimo, da ima flip-flop asinhroni reset, ker se izhod spremeni asinhrono, ne glede na uro. Izhod ostane na 0, dokler je ta signal aktiven, ko gre nazaj na 0, pa začne delovati kot običajno in izhod se lahko spremeni na 1 šele ob naslednji prednji fronti ure.



Slika 5.8: Simbol in časovni diagram registra s signalom reset.

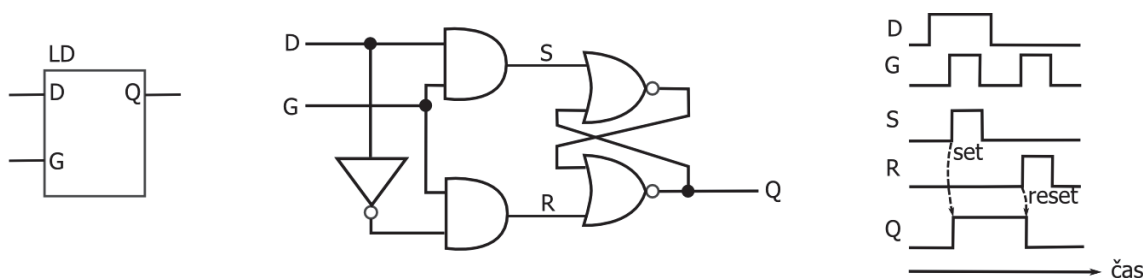
5.3 Izvedba flip-flopov



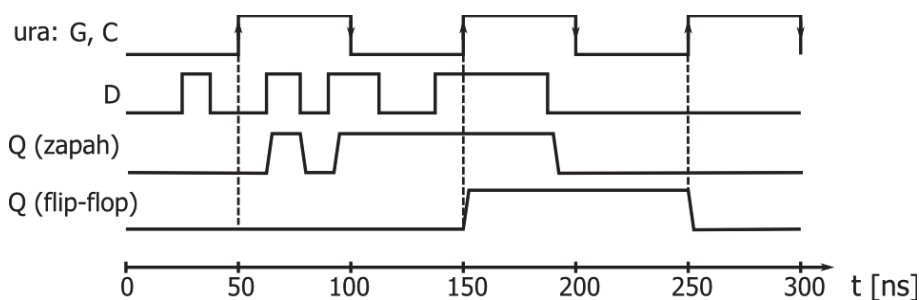
Za izvedbo so najenostavnejši asinhroni pomnilni gradniki oz. zapahi. Zapah z oznako SR ima dva kontrolna vhoda: *S* (angl. Set) in *R* (angl. Reset). Ob aktivnem signalu *S* se izhod *Q* postavi na 1, ob aktivnem *R* pa se postavi na 0. Kadar sta oba kontrolna signala na 0, izhod ohranja zadnjo vrednost. Primer: začetno stanje izhoda *Q* naj bo 0;

- $S=1$ in $R=0$ povzroči, da gre izhod na 1;
- $S=0$ in $R=0$ ohranja izhod na 1;
- $S=0$ in $R=1$ povzroči, da gre izhod na 0;
- $S=0$ in $R=0$ ohranja izhod na 0.

Kadar sta oba vhoda zapaha na 1, izhod ni stabilen in preklaplja med logično 0 in 1. Osnovni stabilen gradnik je zapah D, ki ima podatkovni vhod D , kontrolni vhod (ali vrata) za omogočanje G (angl. Gate) 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.

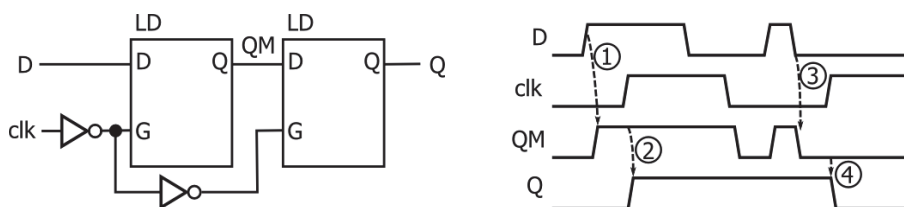


Slika 5.9: Zapah D: simbol, logična shema in časovni diagram.



Slika 5.10: Primerjava časovnega poteka izhoda pri zapahu D in flip-flopu.

Podatkovni flip-flop lahko naredimo s povezavo dveh zapahov. 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.

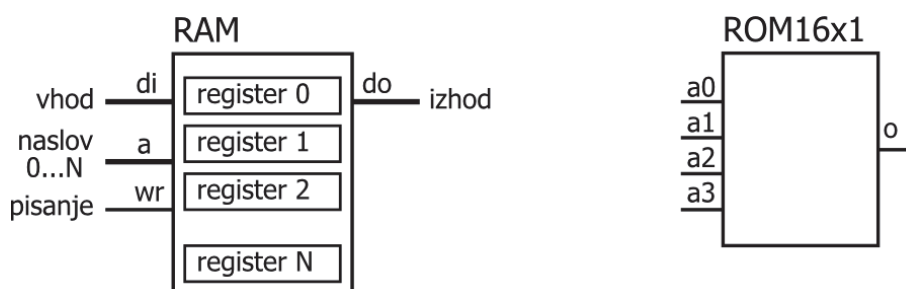


Slika 5.11: Izvedba flip-flopa D iz dveh zapahov.

5.4 Pomnilnik



Pomnilniki so elementi za shranjevanje podatkov in ukazov, ki jih npr. obdelujejo računalniški digitalni sistemi. Pomnilnik shrani več podatkovnih besed, vendar v posameznem ciklu zapisujemo ali beremo le eno besedo naenkrat. Elektronske polprevodniške pomnilnike označujemo s kratico RAM (angl. Random Access Memory), kar pomeni da imajo enak čas za dostop do naključno izbrane pomnilniške besede. Oznaka izhaja iz primerjave s pomnilniki, pri katerih so besede zapisane na magnetnem traku in je čas dostopa do besede odvisen od trenutnega položaja elektromagnetne glave na traku.



Slika 5.12: Blokovna shema pomnilnika RAM in ROM.

Signale na priključkih pomnilnika RAM razdelimo na:

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

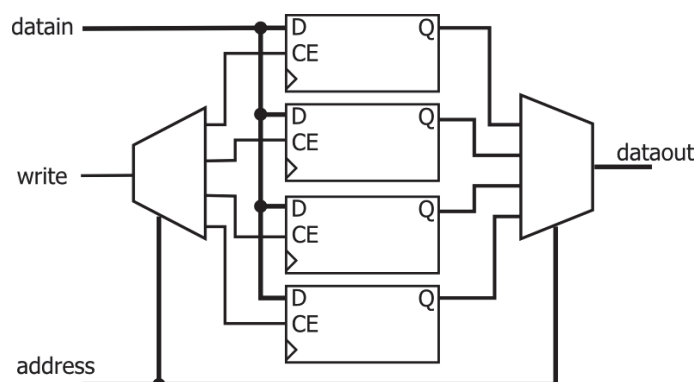
Pomnilne celice si lahko predstavljamo kot registre. Vsak izmed registrov ima svoj naslov v obliki binarnega števila med 0 in N . Na naslovno vodilo postavimo vrednost, s katero izberemo enega izmed registrov, s krmilnimi signali pa določimo, ali bomo v register naložili novo besedo ali pa bomo brali iz registra.

Pomnilniki z oznako ROM (angl. Read Only Memory) imajo vnaprej zapisano vsebino, ki jo preberemo tako, da nastavimo naslovne signale. Slika 5.12 prikazuje simbol pomnilnika ROM s štirimi naslovnimi signali in enim izhodom. Takšen pomnilnik, v katerega lahko shranimo 16 bitov, najdemo v celicah programirljivega vezja.

5.5 Izdelava pomnilnikov



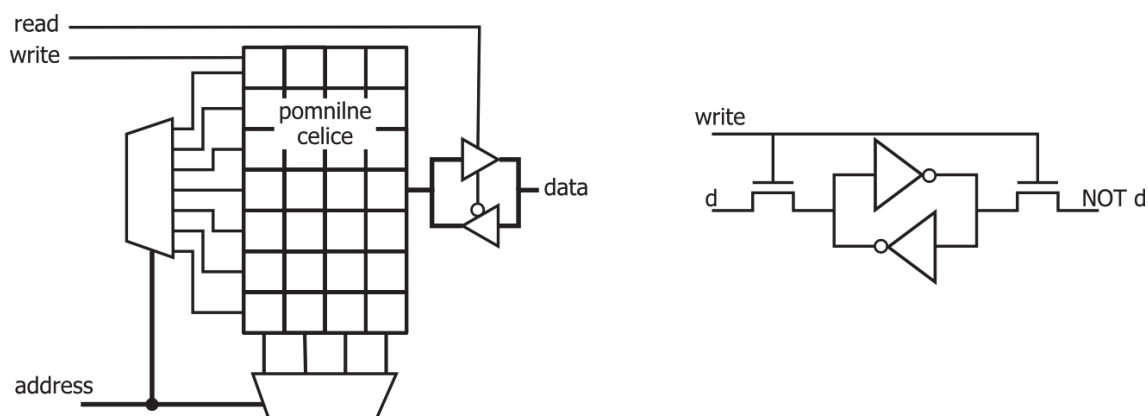
Pomnilnik je mogoče sestaviti iz elementov, ki smo jih do sedaj spoznali: registrov, binarnega dekodirnika in izbiralnika. Registri imajo podatkovne vhode vezane skupaj. Vsakemu registru dodelimo naslov *address* in prek binarnega dekodirnika omogočimo vpis besede v naslovljeni



Slika 5.13: Izvedba pomnilnika RAM iz registrov.

register, kadar je aktiven krmilni signal *write*. Branje poteka tako, da nastavimo naslov, in izbrana beseda se prek izbiralnika prenese na podatkovni izhod.

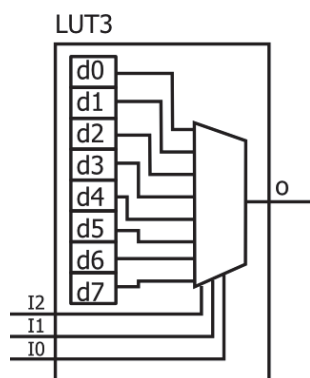
Pomnilnik iz registrov potrebuje za shranjevanje besed veliko elektronskih elementov, ker imajo registri iz flip-flopov D kompleksno zgradbo. V praksi zato uporabljamo učinkovitejše in cenejše pomnilnike, sestavljene iz matrike pomnilnih celic. Slika 5.14 prikazuje zgradbo statičnega pomnilnika RAM in posamezne pomnilne celice.



Slika 5.14: 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 tranzistorjev za vpis nove vrednosti. Za vsak bit potrebujemo skupaj 6 tranzistorjev, kar je precej manj kot pri izvedbi s flip-flopom D. Še učinkovitejši so dinamični pomnilniki, ki hranijo zapis v enem samem tranzistorju, vendar imajo zahtevnejšo krmilno logiko.

Pomnilnik ROM lahko uporabimo kot splošen gradnik za izdelavo kombinacijskih vezij, podobno kot izbiralnik. Vsebina pomnilnika mora biti enaka desni strani pravilnostne tabele kombinacijskega 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 3 do 6 naslovnih vhodov in en izhod.

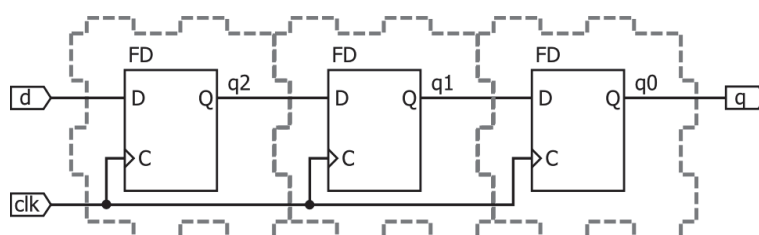


Slika 5.15: Vpogledna tabela LUT3 s tremi naslovnimi vhodi.

5.6 Sekvenčna vezja

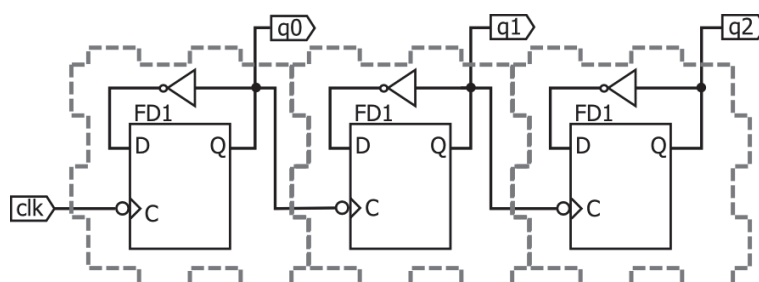


Povezovanje digitalnih gradnikov lahko primerjamo z zlaganjem kock. Simboli digitalnih gradnikov imajo vhodne signale na levi in izhodne na desni strani. *Zaporedno vezavo* naredimo tako, da postavimo več gradnikov v vrsto in izhodne signale povežemo z vhodnimi.



Slika 5.16: Zaporedna vezava podatkovnih flip-flopov.

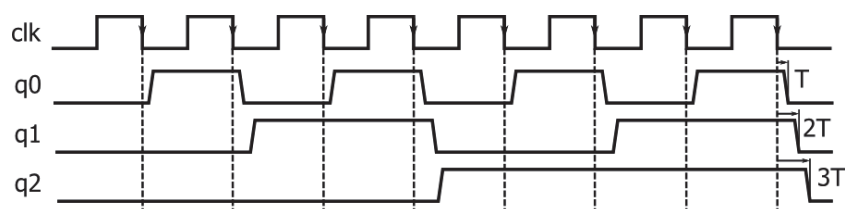
Značilnost zaporedne vezave je zakasnitev pri prenosu podatkov skozi vezje. Kontrolni signali so vezani na vse gradnike (npr. ura), podatki pa prehajajo zaporedno čez gradnike in so zakasnjeni. Pomikalni register s slike 5.16 ima zakasnitev med vhodom in izhodom 3 urne cikle.



Slika 5.17: Zgradba 3-bitnega serijskega števca.

Iz zaporedno vezanih preklonnih flip-flopov naredimo serijski števec, kot prikazuje slika 5.17. 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 zakasnjena 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 ($3T$).

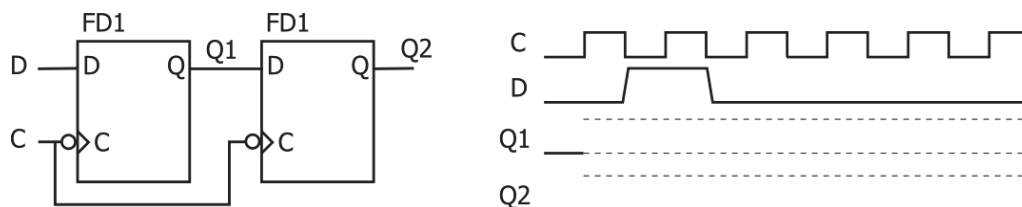


Slika 5.18: Časovni potek signalov 3-bitnega serijskega števca.

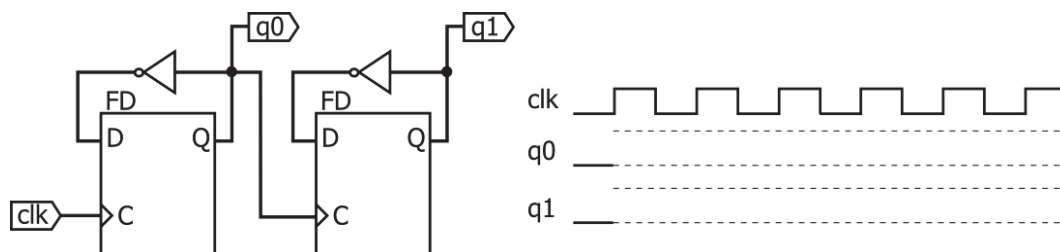
Števec pogosto uporabljamo kot generator zaporedja binarnih vrednosti. Če opazujemo vse izhode hkrati, nam različne zakasnitve preklonov povzročajo težave pri dekodiranju vrednosti in omejujejo najvišjo frekvenco delovanja števca. Zaradi tega števce raje načrtujemo v obliki sinhronega vezja, kjer so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

Naloge

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



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



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