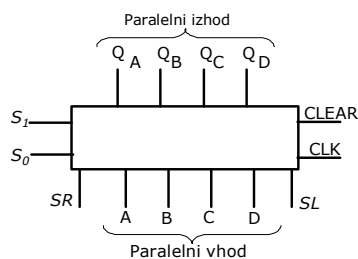


RAZVOJ DIGITALNIH SISTEMOV

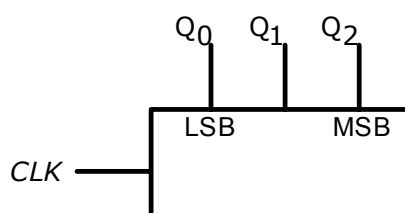
2. kolokvij
17.1. 2019

1. Realizirajte JK-flip flop z uporabo T flip flopa in logičnih vrat.
2. Z uporabo D flip-flopov in izbiralnikov 4/1 prikažite sintezo univerzalnega 4-bitnega pomikalnega registra, ki ima dva funkcijska vhoda S_0 in S_1 in opravlja funkcije po spodnji tabeli. Register ima tudi zaporedna vhoda za pomik v levo (SL – serial left), pomik v desno (SR – serial right) in asinhroni vhod za brisanje CLEAR (aktiven nizek).



| S_1 | S_0 | <i>funkcija</i> |
|-------|-------|-------------------------------|
| 0 | 0 | drži stanje |
| 0 | 1 | pomik vsebine eno mesto desno |
| 1 | 0 | pomik vsebine eno mesto levo |
| 1 | 1 | nalaga vsebino z vhodov ABCD |

3. Prikažite sintezo sinhronnega 3-bitnega števca navzdol z uporabo T flip-flopov: Zapišite tabelo prehajanja stanj in določite enačbe flip-flopov ter vezje narišite. Imena signalov so razvidna iz spodnje slike.



4. Narišite diagram stanj za Mealyev avtomat končnih stanj, ki ima vhod w in izhod z . Avtomat končnih stanj postavi izhod $z='1'$, ko se na vhodu pojavi zaporedje **110** ali **101**, sicer je $z='0'$. Prekrivanje vzorcev je dovoljeno.

Rešitev 1. naloge

Za vezje JK-FF narišemo pravilnostno tabelo, pri čemer na vhodni strani zberemo vhode J, K in trenutno stanje $Q(t)$, na izhodni pa naslednje stanje $Q(t+1)$. JK-FF opravlja štiri funkcije (HOLD, RESET, SET, INVERT) glede na kombinacijo vhodnih signalov, medtem ko T-FF opravlja samo dve (HOLD, INVERT).

| J | K | $Q(t)$ | $Q(t+1)$ | T | funkcija T | funkcija JK |
|-----|-----|--------|----------|-----|--------------|---------------|
| 0 | 0 | 0 | 0 | 0 | HOLD | HOLD |
| 0 | 0 | 1 | 1 | 0 | HOLD | HOLD |
| 0 | 1 | 0 | 0 | 0 | HOLD | RESET |
| 0 | 1 | 1 | 0 | 1 | INVERT | RESET |
| 1 | 0 | 0 | 1 | 1 | INVERT | SET |
| 1 | 0 | 1 | 1 | 0 | HOLD | SET |
| 1 | 1 | 0 | 1 | 1 | INVERT | INVERT |
| 1 | 1 | 1 | 0 | 1 | INVERT | INVERT |

Iz tabele narišemo Veitchev diagram za vhod D v odvisnosti od vhodov J, K in trenutnega stanja $Q(t)$.

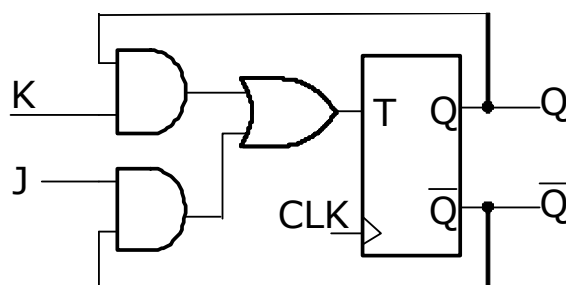
T :

Dobljeni diagram minimiziramo in zapišemo enačbo vhoda T:

| | | | | | |
|-----|--|--------|---|---|---|
| | | J | | | |
| K | | 1 | 1 | 1 | 0 |
| | | 1 | 0 | 0 | 0 |
| | | $Q(t)$ | | | |

$$T = J \cdot \overline{Q(t)} + K \cdot Q(t)$$

Vezje narišemo:



Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk. Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete. Rezultati bodo objavljeni na domači strani predmeta.

Rešitev 2. naloge:

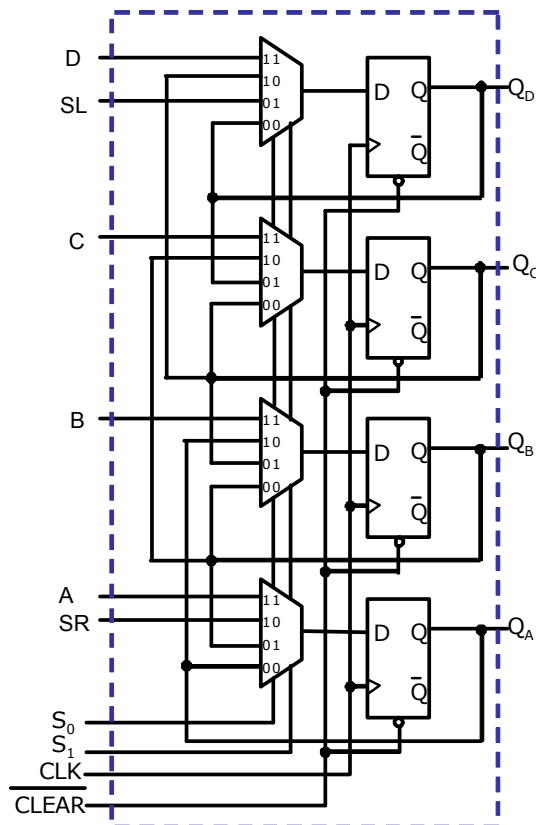
Vsako od operacij izpišemo v pravilnostno tabelo v kateri združimo funkcijska bita S_1 in S_0 in trenutno stanje na i -tem mestu registra $Q_i(t)$. Mesta registra od leve proti desni so $Q_i = (Q_A, Q_B, Q_C, Q_D)$. Realizacija z D flip-flopi nam analizo močno poenostavi, zaradi enačbe D flip-flopa: $D = Q(t+1)$.

Tabela 1: Prehajanje stanj univerzalnega registra.

| S_1 | S_0 | $Q_i(t+1)$ | funkcija |
|-------|-------|--------------|----------|
| 0 | 0 | $Q_i(t)$ | HOLD |
| 0 | 1 | $Q_{i+1}(t)$ | LSR |
| 1 | 0 | $Q_{i-1}(t)$ | LSL |
| 1 | 1 | x_i | LOAD |

Iz poenostavljene tabele prehajanja stanj univerzalnega registra sestavimo realizacijo, ki bo vključevala izbiralnike MUX 4/1 in D-FF.

Na naslovna vhoda vseh MUX 4/1 vodimo funkcijska signala S_1 in S_0 . Potem na vsakem podatkovnem vhodu realiziramo ustrezno funkcijo.



Stanje $S_1S_0 = "00"$ pomeni držanje stanja (*HOLD*), torej bodo trenutne vrednosti D-FF ohranile vrednost $Q_i(t+1) = Q_i(t)$. Na sliki to realiziramo tako, da vodimo izhod D-FF nazaj na vhod pri podatkovnem vhodu 00. Stanje $S_1S_0 = "01"$ pomeni pomik desno (*LSR* – ang. logic shift right), torej bodo D-FF pomaknili vsebino eno mesto desno. Pomik desno pomeni, da na mesto skrajno levega bita vpišemo vrednost zaporednega vhoda SR, nato Q_A vodimo na vhod Q_B in tako do skrajno desnega bita. Stanje $S_1S_0 = "10"$ pomeni pomik levo (*SHL* – ang. shift left), torej bodo D-FF pomaknili vsebino eno mesto levo. Pomik levo pomeni, da na mesto skrajno desnega bita vpišemo vrednost zaporednega vhoda SL, nato Q_D vodimo na vhod Q_C in tako do skrajno levega bita. $S_1S_0 = "11"$ pomeni vzporedno nalaganje z vhodov (*LOAD*) $Q_D(t+1) = D$, $Q_C(t+1) = C$, $Q_B(t+1) = B$, $Q_A(t+1) = A$. Na tabeli smo i -ti vhod za vzporedno nalaganje označili kot $x_i = (A, B, C, D)$.

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk. Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete. Rezultati bodo objavljeni na domači strani predmeta.

Rešitev 3. naloge:

Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števca:

| Trenutno stanje | | | Naslednje stanje | | | Enačbe FF | | |
|-----------------|----------------|----------------|------------------|----------------|----------------|----------------|----------------|----------------|
| Q ₂ | Q ₁ | Q ₀ | Q ₂ | Q ₁ | Q ₀ | T ₂ | T ₁ | T ₀ |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

Normalna analiza bi zahtevala, da narišemo Veitch–eve diagrame za tri spremenljivke za vsak vhod T–FF, vendar ker so T–FF po svoji naravi primerni za realizacijo števec, so praviloma njihove vhodne enačbe zelo enostavne. Iz tabele prehajanja stanj števca določimo enačbe T–FF:

Iz stolpca T₀ se vidi:

$$T_0 = 1$$

Z opazovanjem stolpcev trenutnega stanja določimo T₁:

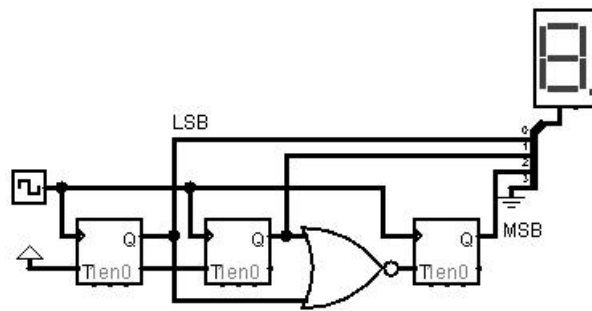
$$T_1 = \overline{Q_0}$$

Podobno lahko določimo T₂:

$$T_2 = \overline{Q_0} \cdot \overline{Q_1} = \overline{Q_0 + Q_1}$$

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk. Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete. Rezultati bodo objavljeni na domači strani predmeta.

Opis delovanja in vezje števca je v predlogah vaj na domači strani predmeta v imeniku Logisim\counter\ counter_7_0_using_T_FF.circ:



Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk. Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete. Rezultati bodo objavljeni na domači strani predmeta.

Rešitev 4. naloge:

Naloga zahteva realizacijo z Mealy–evim tipom avtomata. Zapišemo začetno stanje A, v katerem ostajamo toliko časa, dokler se ne začne ena od sekvenc, ki ju zaznavamo. Obe sekvenci se začneta z '1', zato v stanje B preidemo, ko je na vhodu prva '1'. V stanju B ne moremo ostati, saj se na vhodu lahko pojavi '0' ali '1' – v obeh primerih gre za del zaznavanega zaporedja "10X" ali "11X". Iz stanja B preidemo v stanje C, če se vmes pojavi '1', tako da v tem stanju pomeni detekcijo sekvence "11X", v stanje D pa preidemo če se pojavi na vhodu '0', kar pomeni detekcijo sekvence "10X".

Prekrivanje zaporedij: Če se v stanju C pojavi '1' na vhodu, potem gre za sekvenco "111" na vhodu – kar še vedno pomeni, da ostajamo v stanju C, saj je prekrivanje vzorcev dovoljeno. Drugače se diagram obnaša, ko smo v stanju D in pride na vhod še ena '0' – takrat smo imeli na vhodu sekvenco "100", tako da se moramo vrniti v stanje A, saj se nobena od zaznavanih sekvenc ne začneja z '0'.

