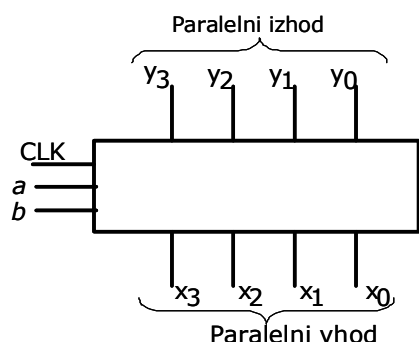


RAZVOJ DIGITALNIH SISTEMOV

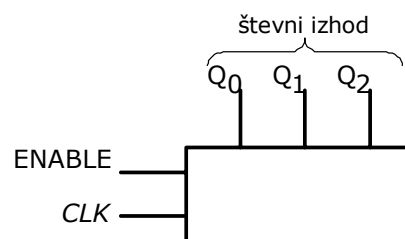
2. kolokvij
18.1. 2018

1. Realizirajte JK-flip flop z uporabo D- flip flopa in logičnih vrat.
2. S pomočjo univerzalnih logičnih modulov (ULM) realizirajte univerzalni register, ki ima dva funkcijska vhoda a in b in opravlja funkcije po spodnji tabeli:



| a | b | <i>funkcija</i> |
|-----|-----|--|
| 0 | 0 | briše vsebino registra (CLEAR) |
| 0 | 1 | rotira vsebino eno mesto desno (ROR) |
| 1 | 0 | negira vsebino (CPL) |
| 1 | 1 | vpiše vsebino s paralelnega vhoda (LOAD) |

3. Prikažite sintezo 3-bitnega sinhronnega števca navzgor z omogočanjem štetja (ENABLE) z uporabo T flip-flopov in logičnih vrat.
Če je ENABLE='1', števec šteje, če je ENABLE ='0' števec stoji (ohranja stanje).



4. Narišite diagram prehajanja stanj Mealy-evega avtomata končnih stanj, ki ima vhod w in izhod z . Avtomat končnih stanj postavi izhod $z=1$, ko se na vhodu pojavi zaporedje **101**, sicer je $z=0$. Prekrivanje vzorcev je dovoljeno.

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 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 D-FF opravlja samo dve (SET, RESET).

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

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

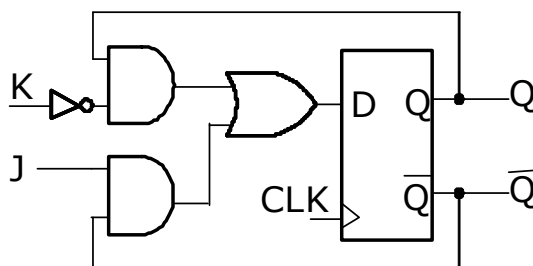
$D:$

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

Dobljeni diagram minimiziramo in zapišemo enačbo vhoda D:

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

Vezje narišemo:



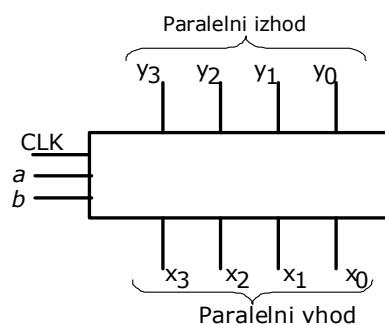
Rešitev 2. naloge:

Naloga zahteva realizacijo univerzalnega registra s funkcijami

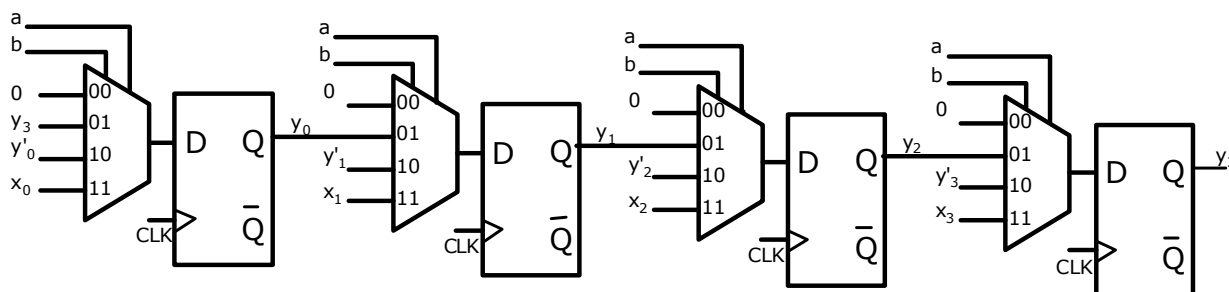
| a | b | <i>funkcija</i> |
|-----|-----|--|
| 0 | 0 | briše vsebino registra (CLEAR) |
| 0 | 1 | rotira vsebino eno mesto desno (ROR) |
| 1 | 0 | negira vsebino (CPL) |
| 1 | 1 | vpiše vsebino s paralelnega vhoda (LOAD) |

Vsako od operacij izpišemo v pravilnostno tabelo v kateri združimo funkcijska bita a , b in trenutno stanje na i -tem mestu registra $y_i(t)$. Realizacija z D flip-flopi nam analizo močno poenostavi, zaradi enačbe D flip-flopa: $D = y_{i+1}(t)$

| a | b | $y_i(t)$ | $y_i(t+1)$ | D |
|-----|-----|----------|--------------|--------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | $y_{i-1}(t)$ | $y_{i-1}(t)$ |
| 0 | 1 | 1 | $y_{i-1}(t)$ | $y_{i-1}(t)$ |
| 1 | 0 | 0 | y_i' | y_i' |
| 1 | 0 | 1 | y_i' | y_i' |
| 1 | 1 | 0 | x_i | x_i |
| 1 | 1 | 1 | x_i | x_i |



Register izvaja rotacijo, torej nima serijskega vhoda in izhoda, ampak LSB bit y_3 vodimo na MSB bit y_0 . Naloga zahteva realizacijo z 4/1 izbiralniki, s katerimi ločimo 4 operacije registra.



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:

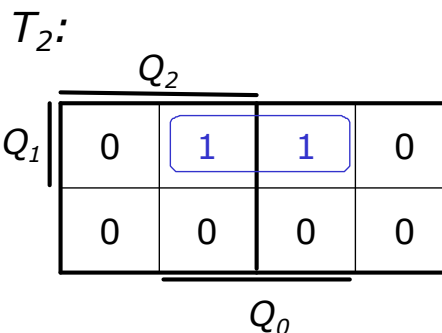
| trenutno stanje | | | naslednje stanje | | | T-FF | | |
|-----------------|----------------|----------------|------------------|----------------|----------------|----------------|----------------|----------------|
| Q ₂ | Q ₁ | Q ₀ | Q ₂ | Q ₁ | Q ₀ | T ₂ | T ₁ | T ₀ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Iz tabele prehajanja stanj števca določimo enačbe T-FF:

Za T₀ se iz tabele vidi T₀ = 1

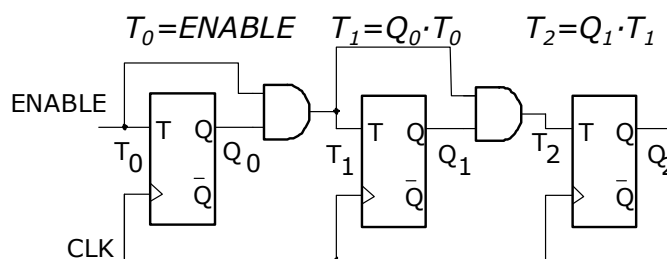
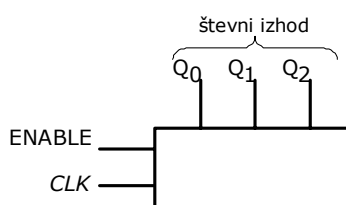
Za T₁ se iz tabele vidi T₁ = Q₀ · 1 = Q₀ · T₀

Za T₂ narišemo Veitchev diagram



$$T_2 = Q_0 \cdot Q_1 = T_1 \cdot Q_1$$

Naloga pravi, da moramo dodati še vhod za omogočanje štetja (ENABLE). Če prvemu T-FF postavimo T₀ = '0' namesto T₀ = '1', flip-flopi števca ne bodo štel, ampak bodo ohranjali stanje. Torej, če na vhod T₀ postavimo zunanji signal ENABLE, števec ne bo štel, ampak ohranjal stanje, če bo ENABLE = '0'. V verigi sinhronnega števca so namreč vezani vsi T-FF tako, da so odvisni od prvega T-FF. Do istega sklepa bi prišli, če bi risali Veitch-eve diagrame za 4 spremenljivke (ENABLE, Q₂, Q₁, Q₀).



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:

Mealy-ev avtomat začnemo risati v začetnem stanju A. Za vsako stanje imamo dve možnosti, saj je vhod (w) samo eden. Vezje ostaja v stanju A (začetno stanje) toliko časa, dokler ne pride na vhod prva '1' zaporedja.

Takrat preide avtomat v stanje B (stanje prva enica), kjer ostane dokler je na vhodu '1'.

Ko se na vhodu pojavi '0', preide v stanje C (stanje enka, ničla), od koder sta dve možnosti:

- Če se v stanju C na vhodu pojavi '0', moramo do sedaj zaznano zaporedje "10X" zavreči, zato se vrnemo v stanje A (začetno).
- Če se v stanju C na vhodu pojavi '1', avtomat postavi izhod na '1' in se vrne v stanje B (stanje prva enica), s čimer omogočimo prekrivanje zaporedij.

