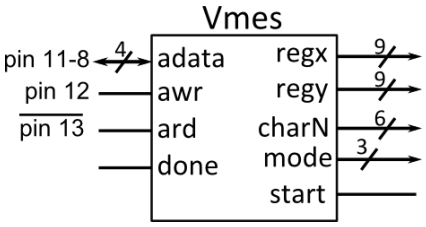
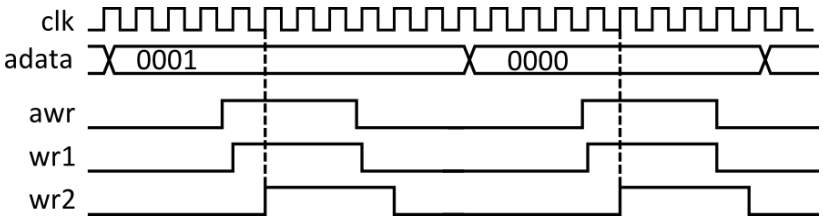
7. Vaja: Vmesnik

Naredi osnovni vmesnik, ki bere ukaze in nastavitve registrov preko 4-bitnega vodila vezanega na vrata mikrokrmilniškega sistema Arduino.

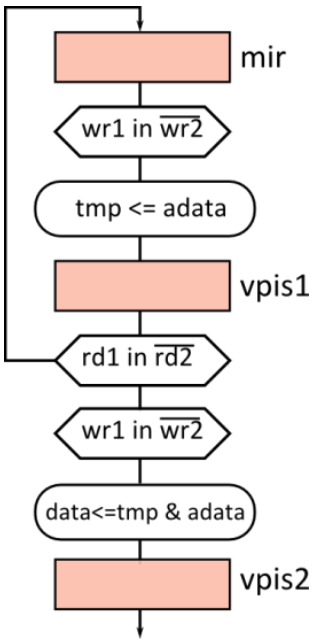




7.1 Komunikacija z Arduinom

Komunikacija poteka asinhrono preko 4-bitnega dvosmernega podatkovnega vodila in dveh krmilnih signalov: **awr** in **ard**. Ukazi ali podatki se vedno prenašajo v obliki dveh zaporednih 4-bitnih prenosnih ciklov. V vsakem ciklu procesor najprej postavi podatek na vodilo, potem pa naredi impulz na signalu **awr**.

Vezje naj vzorči krmilne signale z dvema zaporednima flip-flopoma. Iz kombinacije vzorčenih signalov vezje zazna impulz na krmilnem signalu in določi trenutek za shranjevanje podatka. Primer: signal wr1 je enkrat vzorčen signal **awr**, signal wr2 pa je zakasnjen še za en urni cikel. Ob kombinaciji wr1='1' in wr2='0' sinhrono določimo prvo fronto impulza na signalu **awr** in shranimo podatek iz vodila v notranji register.



Branje statusa grafičnega vmesnika poteka ob impulzu na signalu **ard**, ki ga prav tako dvakrat vzorčimo. Ob vsaki zahtevi za branje naj se avtomat vmesnika postavi v začetno (mirovno) stanje.

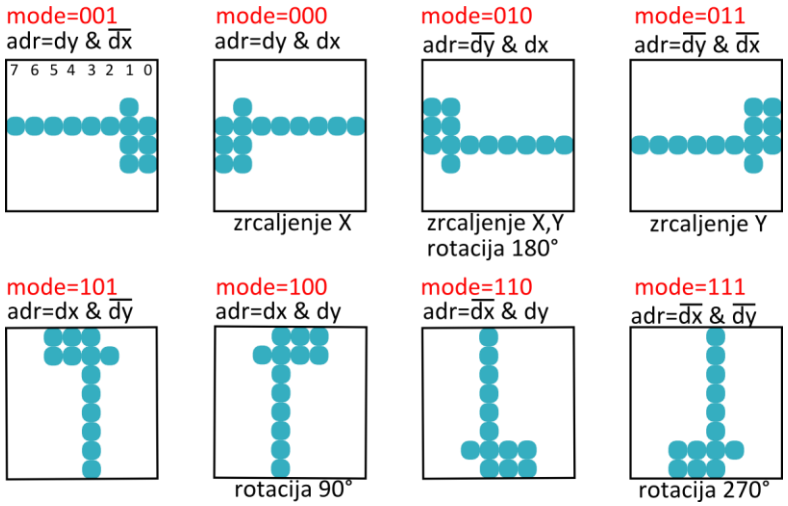
Vmesnik naredimo kot sekvenčno vezje z algoritmičnim diagramom stanj. V stanju vpis2 se odločamo, kaj narediti s sprejetim podatkom. Primer:

- Če je data = 0001, smo prejeli ukaz SetRegX, ki si ga zapomnimo, nastavimo najvišji bit 9-bitnega registra regx(8) <= data(0), in počakamo naslednji podatek.
- Če je data = 0010, smo prejeli ukaz SetRegY…
- Če je data = 1000, smo prejeli ukaz DrawChar, si ga zapomnimo in nastavimo register mode…
- Če smo v prejšnjem ciklu prejeli ukaz SetRegX, potem postavimo še preostale bite registra: regx(7 downto 0) <= data.
- Če smo v prejšnjem ciklu prejeli ukaz DrawChar, potem nastavimo kodo znaka: charN <= data(5 downto 0) in naredimo impulz na izhodu: start<='1'
- …

7.2 Ukazi

Osnovna izvedba vmesnika naj pozna tri ukaze: nastavljanje registrov regx in regx in prikaz znaka (c = charN) v izbranem načinu (m = mode), ki določa rotacijo in zrcaljenje prikazanega znaka.

ime ukaza	**koda ukaza**		**podatek**	
SetRegX	0 0 0 1	- - - r(8)	r(7) r(6) r(5) r(4)	r(3) r(2) r(1) r(0)
SetRegY	0 0 1 0	- - - r(8)	r(7) r(6) r(5) r(4)	r(3) r(2) r(1) r(0)
DrawChar	1 0 0 0	- m(2) m(1) m(0)	- - c(5) c(4)	c(3) c(2) c(1) c(0)



7.3 Programske funkcije za prikaz besedila

Naredi dve osnovni funkciji za prikaz besedila iz znakov:

- GrafXY(int x, int y) naj določi začetne koordinate za prikaz besedila (ukaza: SetRegX in SetRegY)
- GrafPuts(char *s) naj prikaže znakovni niz (zaporedoma pošilja ukaz: DrawChar(mode=001)).

V globalni funkciji setup() poskrbi za pravilne nastavitve vhodno-izhodnih vrat: priključek 13 naj bo izhod s privzeto vrednostjo 1, ki določa smer za pisanje podatkov, priključek 12 pa izhod s privzeto vrednostjo 0.

Pri pisanju na vrata uporabljaj ukaze za neposreden dostop do registrov vrat, ki delajo hitreje kot vgrajene funkcije.

Funkcija za prikaz besedila iz znakovnega niza naj pred pošiljanjem posameznega znaka preoblikuje kodo, ker pozna grafični vmesnik le 64 različnih znakov, ki se začnejo s presledkom. Funkcija naj male črke spremeni v velike in od kode ASCII naj odšteje vrednost 32, da dobi kodo znaka (c).