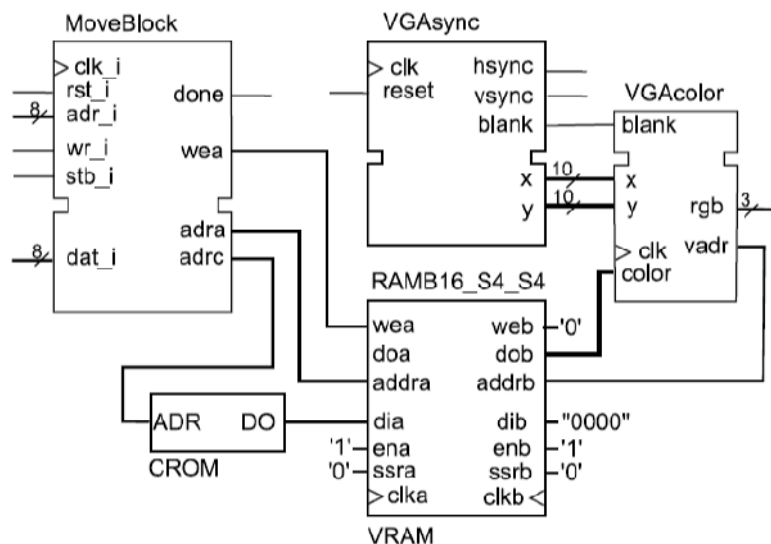


Prikazovalna enota za VGA monitor

3. del: Enota za premikanje podatkov

Naredi podvezje **MoveBlock**, ki izvaja premikanje podatkov med znakovnim pomnilnikom CROM in video pomnilnikom VRAM. Za video pomnilnik bomo vzeli narejeno strojno komponento (angl. primitive) BLOCK RAM, v izvedbi z dvema vrati in velikostjo 4096 x 4 bite. Vrata A bomo uporabili za pisanje, vrata B pa naj bodo vezana na podvezje **VGAclock**, ki izvaja branje in prikaz na monitorju.

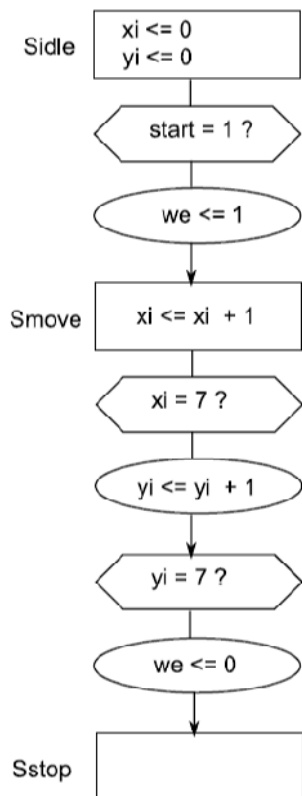


Vhodi vezja **MoveBlock** naj bodo prilagojeni za priključitev na vodilo WishBone.

Vezje **MoveBlock** ima 4 izhode:

- **done**, ki se postavi na '1' ob koncu premikanja bloka podatkov
- **wea** določa vpis na vrata A video pomnilnika
- **adra** je 12 bitni naslov za vpis podatkov na vrata A video pomnilnika
- **adrc** je 12 bitni naslov za branje iz znakovnega pomn.

3 a) avtomat za premikanje podatkov



Grafične kartice izvajajo poleg prikazovanja slike tudi naloge, ki pohitrijo generiranje slike v pomnilniku. Ena izmed osnovnih nalog je premikanje podatkov. Uporabimo jo npr. za izpis besedila, kjer premikamo posamezne znake iz znakovnega pomnilnika v video pomnilnik. V znakovnem pomnilniku so zapisani znaki v obliki sličic, kjer vsak znak zasede 64 (8x8) zaporednih lokacij. Naslov pomnilnika (adrc) je 12 biten – zgornjih 6 bitov določa znak, spodnjih 6 pa točke znotraj znaka.

Naredi avtomat, ki začne ob pogoju (start='1') premikati podatke iz znakovnega v video pomnilnik. Delovanje avtomata ponazorimo z algoritmičnim diagramom prehajanja stanj. V tem diagramu so stanja predstavljena s pravokotniki, v katerih so stavki, ki se izvršijo v določenem stanju. Pogoji so zapisani z odločitvenimi bloki (šestkotniki), v elipsah pa so stavki, ki se izvršijo ob izpolnjenem pogoju. Avtomat izvaja premikanje podatkov z dvema števcema: yi je števec vrstic, xi pa števec točk – en premik pomeni prenos osmih točk v vsaki od osmih vrstic (skupaj 64 premikov), pri čemer je izhod we postavljen na 1.

Naslov znakovnega pomnilnika je sestavljen iz naslova znaka (6 bitov) in yi in xi.

$$\text{adrc} \leq \text{c} \& \text{yi} \& \text{xi}; \quad \text{-- signal c je 6 bitni register}$$

Naslov video pomnilnika pa dobimo tako, da k osnovnemu naslovu (4-bitni y in 8-bitni x) prištejemo oba inkrementa:

$$\text{adra} \leq (\text{y} + \text{yi}) \& (\text{x} + \text{xi});$$

3 b) registri vodila Wishbone

Naredite proces, ki opisuje registre x (8-bitni register), y (4-bitni register) in c (6-bitni register). Vpis podatkov v registre poteka ob fronti ure in pogoju stb_i = '1' in wr_i = '1' in sicer:

- ob naslovu adr_i = x"04" se v register c vpiše spodnjih 6 bitov iz vodila dat_i in postavi se signal start za zagon avtomata,
- ob naslovu adr_i = x"05" se v register x vpišejo podatki iz dat_i,
- ob naslovu adr_i = x"06" pa se v register y vpišejo spodnji 4 biti iz vodila dat_i.

V tem procesu realizirajte tudi zastavico done, ki se postavi na '1', ko pride avtomat v stanje Sstop. Zastavica naj se avtomatsko pobriše ob vpisu podatka na naslov x"04".

```
p_reg: process(clk_i)
begin
if rising_edge(clk_i) then
start <= '0';

if stb_i='1' and we_i='1' then
if adr_i=X"04" then
c <= dat_i(5 downto 0);
start <= '1';
done <= '0';
...
end process;
```

3 c) prevajanje in simulacija

Delovanje podvezja **MoveBlock** preverite s simulacijo v programu ModelSim s priloženo makro datoteko move.do.

Ko pokaže simulacija pravilno delovanje, se lotite prevajanja celotnega sistema. Najprej naredite manjkajoči model programskega pomnilnika, tako da prevedete zbirniško kodo v programu pBlaze IDE. Program naredi ob vsakem prevajanju novo datoteko z VHDL modelom programskega pomnilnika (prog_rom.vhd). Sedaj lahko naredite sintezo vezja (VGA_Micro) in simulacijo celotnega sistema v programu ModelSim.

