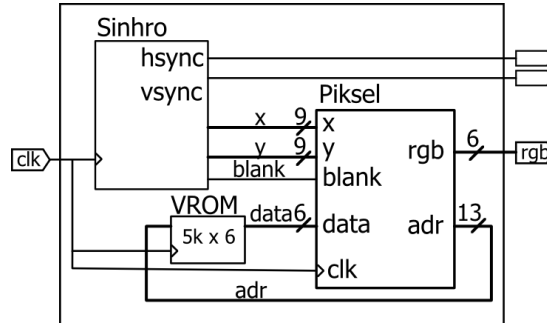


## 5. Vaja: Prikaz slike iz pomnilnika

Na monitorju želimo prikazovati sliko, ki je shranjena v blokovnem pomnilniku znotraj FPGA. Vezje za prikaz slike naj bo sestavljeno iz treh komponent:



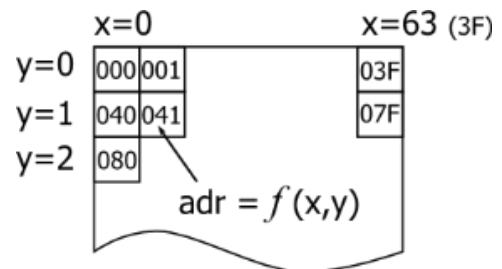
### 5.1 Komponenta Sinhro

Za izdelavo sinhronizacijske komponente uporabi vezje iz vaje 4, ki mu odstrani izhod rgb in dodaj 3 izhode: koordinate x in y ter signal blank. Izhodne koordinate so kar enake vrednosti horizontalnega oz. vertikalnega števca, signal blank pa naj ima vrednost 1, kadar je števec znotraj vidnega dela slike (512 x 480).

### 5.2 Komponenta Piksel

Glavna komponenta, ki določa barvo izhodnih točk, dobi koordinate iz sinhronizacijske komponente in vrednosti točk iz video pomnilnika. V video pomnilniku bo majhna sličica velikosti 64 x 80 točk, ki jo prikažemo na enem koncu vidnega dela izhodne slike. Naloga komponente Piksel je, da bere podatke iz pomnilnika in jih v ustreznih trenutkih prenese na izhod rgb.

Komponenta mora nastaviti naslov pomnilnika glede na trenutne koordinate točk (x in y), kot prikazuje slika. Pri branju podatkov upoštevaj, da ima pomnilnik zakasnitev (latenco) en urni cikel – če nastavimo naslov ob fronti prvega cikla, bomo prebrali podatek ob fronti tretjega cikla !

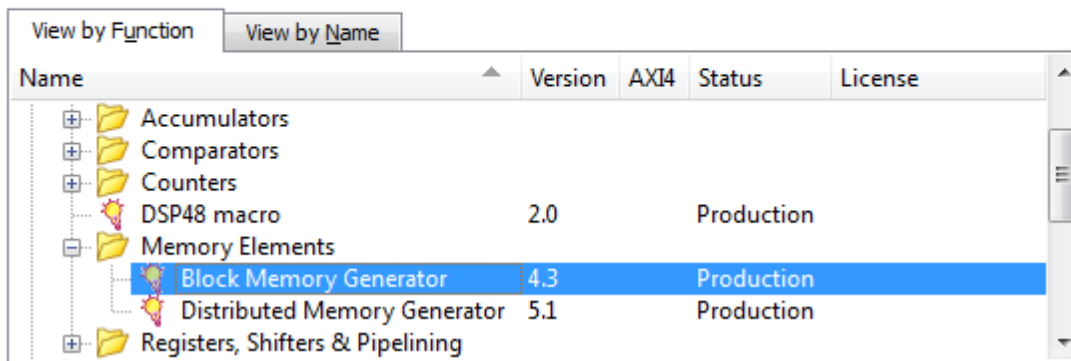


### 5.3 Komponenta VROM

Pomnilnik bomo naredili z uporabo čarovnika v obliki komponente intelektualne lastnine (angl. IP). V programu ISE dodamo novo izvorno datoteko vrste IP CORE z imenom VROM. Uporabimo komponento Block Memory, ki jo poiščemo med pomnilnimi elementi.

## Select IP

Create Coregen or Architecture Wizard IP Core.



Pomnilnik definirajmo kot ROM širine 6 bitov (Data Width) s 5120 lokacijami (Depth). Vsebina pomnilnika je na voljo v tekstovni datoteki slika.coe, ki jo naložimo iz diska. Na koncu kliknemo Generate in program bo v imeniku ipcore\_dir naredil tehnološko datoteko z načrtom pomnilnika in omejitvami (\*.ngc) ter nekaj pomožnih datotek. Najbolj zanimiva je CROM.vho, ki vsebuje deklaracijo in primer povezave komponente v jeziku VHDL.

## 5.4 Celotno vezje

Celotno vezje za prikaz slike iz pomnilnika naredimo kot nov VHDL opis vezja v katerem deklariramo in povežemo vnaprej pripravljene komponente.

**architecture Behavioral of Vga is**

**component** VROM **is**

**port** (clka: **in** std\_logic;

    addra: **in** std\_logic\_vector(12 **downto** 0);

    douta: **out** std\_logic\_vector(5 **downto** 0));

**end component;**

...

**signal** x, y: std\_logic\_vector (8 **downto** 0); -- *notranji signali za povezave komponent*

**signal** blank: ...

**begin**

U3: VROM **port map** (clka => clk, addra => addra, douta => douta);

U2: Pixel **port map** (x => x, y => y, ...