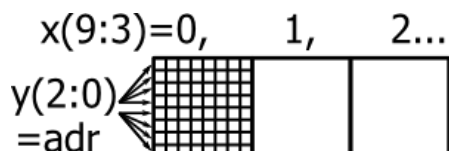


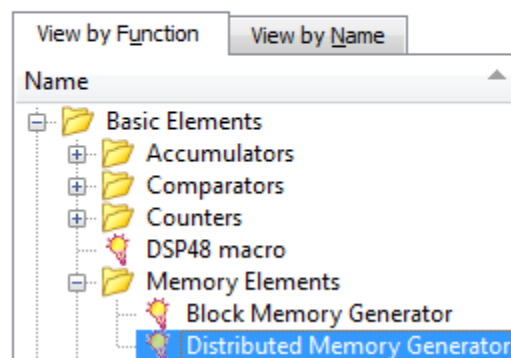
6. Vaja: Prikaz znakov

V grafični krmilnik bomo dodali pomnilnik s 64 sličicami znakov (ASCII kode od 32 do 95) velikosti 8 x 8 točk. Pomnilnik velikosti 512 x 8 bitov bomo naredili z generatorjem intelektualnih komponent. Sličica znaka je zapisana v osmih zaporednih pomnilniških lokacijah.



Select IP

Create Coregen or Architecture Wizard IP Core.



Naslov pomnilnika deklariramo kot 9-bitni standardni vektor. Sestavljen je iz naslova znaka (zgornjih 6 bitov) in spodnjih 3 bitov vertikalnega števec y .

```
adr <= "100001" & std_logic_vector( y(2 downto 0) );
```

Pomnilnik naredimo v programu ISE kot novo izvorno datoteko vrste IP CORE z imenom crom. Najprej izberemo vrsto komponente: **Basic Elements > Memory Elements > Distributed Memory Generator**, nato pa definiramo: vrsto pomnilnika **ROM**, širine 8 bitov (**Data With**) s 512 lokacijami (**Depth**). Vsebina pomnilnika je na voljo v tekstovni datoteki Crom512x8.coe, ki jo naložimo iz diska. Na koncu kliknemo **Generate** in program bo v imeniku ipcore_dir naredil tehnološko datoteko z načrtom pomnilnika, ki ga vključimo v vezje s stavkom port map:

```
ROM: crom port map (a => adr, spo => row);
```

Definirajmo še nekaj notranjih signalov:

- row: std_logic_vector(7 downto 0)
- izpis: std_logic;
- xz: unsigned(6 downto 0) := "0001010"; -- začetna koordinata x
- yz: unsigned(5 downto 0) := "000101"; -- začetna koordinata y

Signal *izpis* določa ali se (x,y) nahaja znotraj sličice znaka s koordinatami (xz, yx):

```
izpis <= '1' when x(9 downto 3) = xz and y(8 downto 3) = yz else '0';
```

Znak narišemo v procesu, kjer imamo že risanje gibljivih sličic. Dodajmo pogoj:

```

elsif izpis='1' then
  if row(to_integer(7 - x(2 downto 0)))= '1' then -- preveri vrstico znaka
    rgb <= "111100";
  else
    rgb <= "000011"; -- ozadje
  ...

```

Preveri delovanje vezja na razvojnem sistemu, nato pa poskusi prikazati par zaporednih znakov.