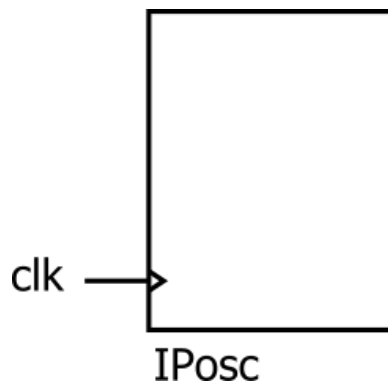


9. vaja: IP-komponenta osciloskop – 1. del

Naredi komponento IP z vhodno logiko digitalnega osciloskopa, ki izvaja skaliranje podatkov, izbiro frekvence vzorčenja ter proženje in zajem podatkov.

9.1 Komponenta IPosc

Naredi novo vezje IPosc, ki naj vključuje med seboj povezani komponenti za skaliranje podatkov in izbiro frekvence vzorčenja. Med priključke vezja dodaj pomnilniško vodilo: 9-bitni vhod **adr** in 9-bitni izhod **dataout** za branje zajetih vzorcev ter enobitni vhodni signal (**s1**) za začetek proženja. Določi na blokovni shemi zunanje priključke komponente:



9.2 Pomnilnik in prožilni avtomat

V arhitekturi vezja IPosc deklariraj podatkovni tip in notranji signal za pomnilnik velikosti 1024 x 9 bitov:

```
type mem is array(0 to 1023) of signed(8 downto 0);
signal ram: mem := (others=>(others=>'0'));
```

Napiši stavek za branje vsebine pomnilnika: `dataout <= ram(to_integer(unsigned(adr)));`

Naredi sekvenčno vezje za zajem vzorcev vhodnih podatkov ob izpolnjenih pogojih za proženje. Vezje naj bo narejeno kot sekvenčni avtomat s stanji: *mir*, *start* in *zajem*. V stanju *mir* čakamo na aktiven signal z vhoda **s1**, ob katerem preklopimo v stanje *start*, kjer bomo opazovali dve zaporedni vrednosti vhodnih podatkov.

Proženje zajema podatkov naj bo povezano s prehodom skaliranih podatkov čez vrednost 0. Pogoj za začetek zajemanja podatkov je, da je trenutna vrednost podatka >0 in predhodna vrednost <=0. V stanju *zajem* naj se podatki shranjujejo v zaporedne naslove pomnilnika ob pogoju **oe='1'** in ob vsakem shranjenem podatku naj se poveča 9-bitni števec (**n**):

```
if st=zajem and oe='1' and n<1023 then
  ram(to_integer(n)) <= d;
  n := n + 1;
end if;
```

Ko pride števec do konca ($n=1023$) naj se sekvenčni stroj vrne v začetko stanje (*mir*), kjer čaka na ponovno aktiviranje proženja. Preizkusi delovanje komponente na simulaciji.