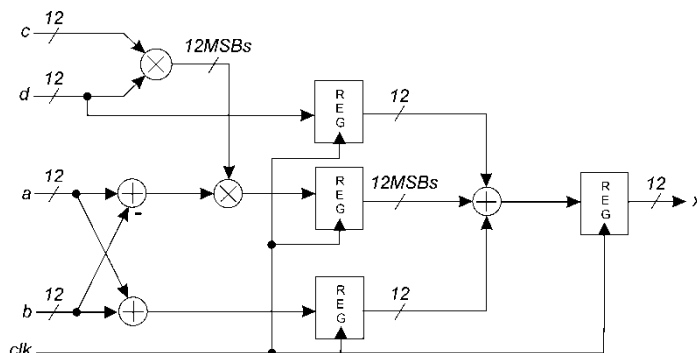


Alati za razvoj digitalnih sustava

2. domaća zadaća

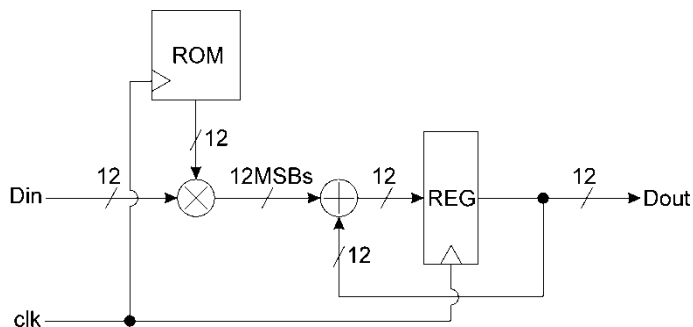
Zadatak 1 (6 boda)



Realizirati sklop prikazan slikom. Ulazi u sklop su **a**, **b**, **c**, **d** i **clk**, a izlaz **x**. Pored ovih signala potrebno je predvidjeti signal za sinkroni reset registara. Pretpostaviti da signali **a**, **b**, **c** i **d** sadrže vrijednosti u dvojnomoj komplementu.

(Napomena: *MSBs* označava *Most Significant Bits* tj. najznačajnije bitove.)

Zadatak 2 (7 boda)



Realizirati sklop prikazan slikom. Ulazi u sklop su 12-bitni podatak **Din**, **clk** i sinkroni reset **rst**, a izlaz je 12-bitni podatak **Dout**. ROM memoriju sa 128 lokacija realizirati korištenjem blok RAM-a. Blok RAM realizirati korištenjem slijednog opisa.

Adresni generator realizirati kao sinkrono brojilo.

Zadatak 3 (10 bodova)

Zadan je RTL model sklopa čiji entitet ima oblik

```
entity filter is
  port( clk    : in  std_logic;
        reset  : in  std_logic;
        Din    : in  std_logic_vector(11 downto 0);
        Dout   : out std_logic_vector(13 downto 0) );
end filter;
```

Uz pretpostavku da je frekvencija takta 50MHz, potrebno je razviti ispitno okruženje koje sadrži pobudne signale. Signal **reset** mora biti aktivan barem prvih 16 perioda takta. Vrijednosti signala **Din** treba predstavljati niz počevši od najveće do najmanje pozitivne vrijednosti. Signal **Din** potrebno je generirati svaki rastući brid takta i to nakon deaktivacije signala **reset**. Napisati komandnu datoteku za pokretanje funkcijske simulacije. Pretpostaviti da su podaci **Din** i **Dout** u formatu **signed**.

Zadatak 4 (10 bodova)

Potrebno je razviti RTL model dvoprístupne sinkrone ROM memorije korištenjem blok RAM memorije. Sadržaj memorije potrebno je zadati u datoteci s imenom **rom.txt**. Memorija mora imati generičku širinu adresne i podatkovne riječi.

Zadatak 5 (6 boda)

Korištenjem isključivo komponente iz biblioteke čiji je opis dan niže realizirati 4/1 multipleksor. Ulazi u sklop su 4 1-bitna signala, **a**, **b**, **c**, **d**, 2-bitni upravljački signal, **s**, a izlazni signal je **y**.

Zadatak 6 (3 boda)

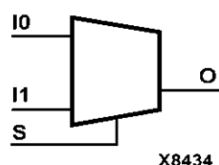
Opisati izvedbu distribuiranog RAM-a. Koje resurse unutar FPGA sklopa troši distribuirani RAM?

Zadatak 7 (3 boda)

Opisati izvedbu množila u Spartan3 FPGA sklopovima. Na koji način se mogu modelirati množila? Kako će sintetizator razlikovati množila sa predznakom i bez predznaka?

MUXF6**2-to-1 Lookup Table Multiplexer with General Output****Architectures Supported**

MUXF6	
Spartan-II, Spartan-IIE	Primitive
Spartan-3	Primitive
Virtex, Virtex-E	Primitive
Virtex-II, Virtex-II Pro, Virtex-II Pro X	Primitive
XC9500, XC9500XV, XC9500XL	No
CoolRunner XPLA3	No
CoolRunner-II	No



MUXF6 provides a multiplexer function in a full Virtex, Virtex-E, Spartan-II, or Spartan-IIE CLB, or one half of a Spartan-3, Virtex-II, Virtex-II Pro, or Virtex-II Pro X CLB (two slices) for creating a function-of-6 lookup table or an 8-to-1 multiplexer in combination with the associated four lookup tables and two MUXF5s. The local outputs (LO) from the two MUXF5s in the CLB are connected to the I0 and I1 inputs of the MUXF6. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The variants, "MUXF6_D" and "MUXF6_L", provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

Inputs			Outputs
S	I0	I1	O
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

Usage

For HDL, this design element can only be instantiated.

VHDL Instantiation Template

```
-- Component Declaration for MUXF6 should be placed
-- after architecture statement but before begin keyword

component MUXF6
    port (O : out STD_ULOGIC;
          I0 : in STD_ULOGIC;
          I1 : in STD_ULOGIC;
          S : in STD_ULOGIC);
end component;
```