TTMER102 - FPGA-alapú hálózati eszközfejlesztés

Elméleti kérdések:

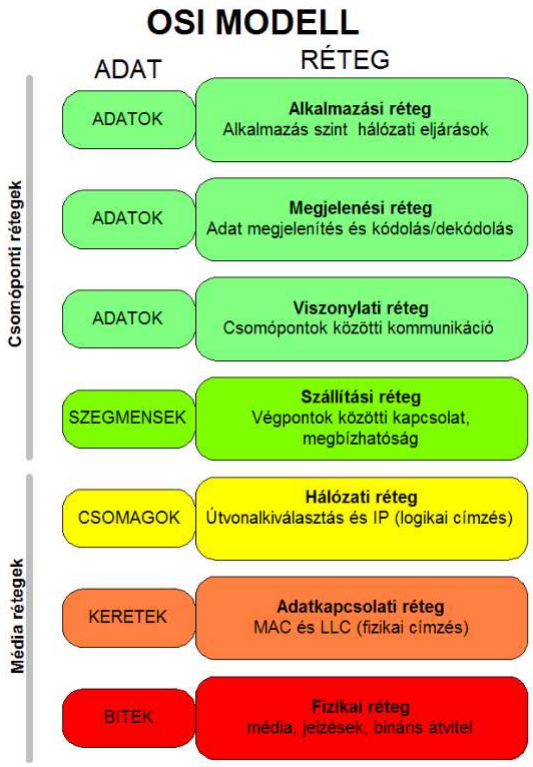
1. **Mi a távközlésben használt switchek funkciója (melyik rétegben, mire használják)?**
   L2, azaz az adatkapcsolati rétegben használják. A MAC címek vizsgálatával képesek közvetlenül a célnak megfelelő portra továbbítani az adott keretet; tekinthetők gyors működésű, többportos hálózati hídnak is. Portok között ütközés nincs, saját sávszélességgel rendelkezik minden port.

2. **Mi a távközlésben használt router funkciója (melyik rétegben, mire használják)?**
   L3, azaz a hálózati rétegben használják. Feladat különböző hálózatok összekapcsolása, az azok közötti adatforgalom irányítása. Pl. otthoni / irodai hálózat és az internet összekapcsolása

3. **Mi a távközlésben használt hub funkciója (melyik rétegben, mire használják)?**
   L2, azaz az adatkapcsolati rétegben használják. Az egyik portján érkező adatokat továbbítja az összes többi portra csatlakozó eszközök felé.

4. **Mi a különbség a switch és router használati célja között?**
   A switch használati célja, hogy a hálózaton belül továbbítsa a csomagokat, ezzel szemben a routert különböző hálózatok összekapcsolására használják.

5. **Rajzolja le OSI model rétegeit és röviden nevezze meg őket!**



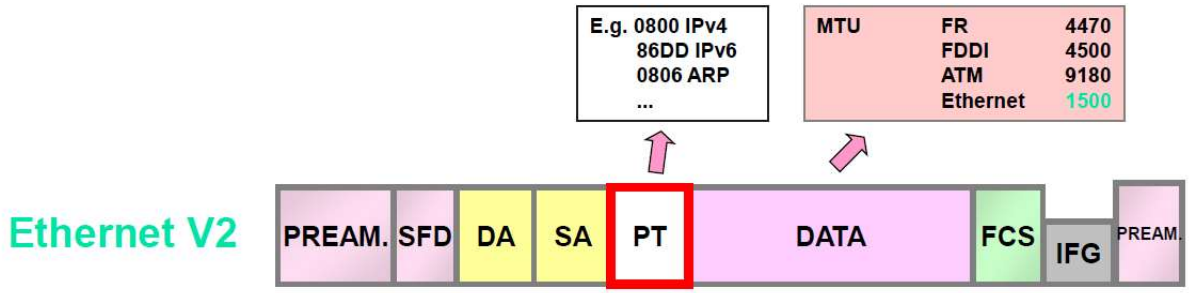
6. **Hány bájt hosszúak a legfontosabb (IPv4,IPv6, TCP, UDP, Eth2) headerek?**
   IPv4: 32 bájt
   IPv6: 36 bájt
   TCP: 20 bájt
   UDP: 28 bájt
   Eth2: 18 bájt

7. **Rajzolja le a IPv4 keretet, nevezze meg a fontosabb mezőket!**

Verzió	Fejléchossz	Típus	Teljes hossz
Azonosító			Flagek + Darabeltolás
Élettartam		**Protokoll**	Fejléc ellenőrző összeg
Forrás cím			
Cél cím			

   Layer-3: Hálózati réteg: Esetünkben ez az IP fejlécet jelenti [**lényeges mezők: Layer-4 protokoll azonosítója, forrás IP cím, cél IP cím**]

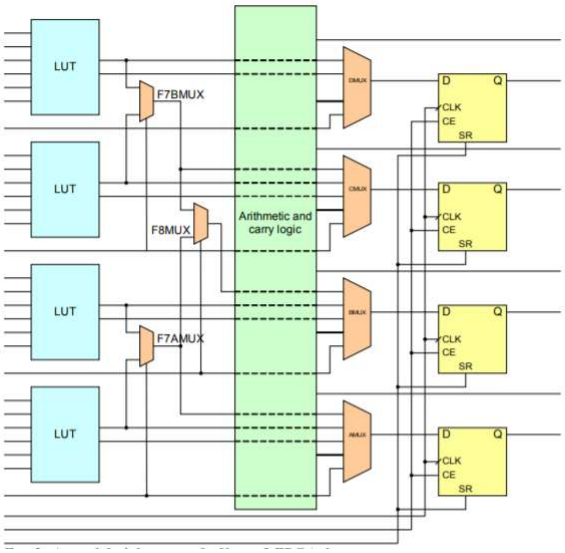
8. **Rajzolja le az Ethernet II keretet és nevezze meg a fontosabb mezőit!**



   Pream.: Preamble
   SFD: Start of Frame Delimiter
   DA: Destination Address
   SA: Source Address
   PT: Protocol Type
   FCS: Frame Check Sequence
   IFG: Inter Frame Gap
   Lényeges mezők: SA, DA, PT, FCS

FPGA kérdések:

9. **Röviden ismertesse a FPGA (Virtex 5) logikai blokkjainak felépítését!**



4x **LUT** (look-up tables)
3x dedikált felhasználó által vezérelhető **multiplexer** kombinációs logika megvalósítására
1x dedikált **aritmetikai logika**
4x **1-bites regiszter** ami flip-flopként és latch-ként is konfigurálható

10. **Nevezzen meg 4 különböző FPGA erőforrást!**
LB/CLB: konfigurálható logikai blokkok
IOB: I/O blokkok
PI: FPGA belső komponensei között programozható összeköttetés
DCM: digitális órajel menedzselő áramkör
BRAM: Blokk-RAM memóriák
MULT/DSP: beágyazott szorzó áramkörök

11. **Mi a különbség a Flip-flop és a latch között?**
A Flip-flop szinkron működésű, azaz a kimenete csak órajel hatására változhat. A latch aszinkron működésű, azaz a kimenet a bemenet változására megváltozhat.

12. **Hogyan tudjuk elkerülni latch-ek keletkezését?**
FPGA programozása esetén: figyelni kell, hogy minden bemeneti kombinációra definiálva legyen a kimenet.

13. **Hasonlítsa össze röviden a szinkron és az aszinkron sorrendi hálózatok tulajdonságait!**
Szinkron: Az állapotváltozás egy engedélyező jel hatására, azzal azonos fázisban zajlik le. Ezt az engedélyező jelet órajelnek, vagy más néven ütemjelnek nevezzük. A kimenet előző állapotától való függést tárolók segítségével valósítják meg.
Aszinkron: Ez a szekvenciális hálózat azon fajtája, amelynél a kimenet előző állapotától való függését visszacsatolással vagy tárolókkal valósítják meg. A bementi jellemző megváltozására a kimenti jellemző azonnal reagál.

VHDL feladat példák:

14. VHDL shiftregiszter aszinkron resettel.

```
1   ......
2   clk, sh, din, rst: in std_logic;
3   dout: out std_logic;
4   ......
5   signal shr: std_logic_vector(15 downto 0);
6   ......
7   process (clk, rst)
8       if (rst='1') then
9           shr <= (others => '0');
10      elsif (clk'event and clk='1') then
11          if (sh='1') then
12              shr <= shr(14 downto 0) & din;
13          end if;
14      end if;
15  end process;
16
17  dout <= shr(15);
```

15. VHDL számláló szinkron resettel.

```
.....
clk, rst, ce, load, dir : in std_logic;
din: in std_logic_vector(7 downto 0),
dout: out std_logic_vector(7 downto 0);
.....
signal cntr_reg : std_logic_vector(7 downto 0);
.....
process (clk)
begin
if (clk'event and clk='1') then
  if (rst='1') then
    cntr_reg <= (others=>'0');
  elsif (ce='1')
    if (load='1') then
      cntr_reg <= din;
    elsif (dir='1') then
      cntr_reg <= cntr_reg - 1;
    else
      cntr_reg <= cntr_reg + 1;
    end if;
  end if;
end if;
end process;

dout <= cntr_reg;
```