

Digitális Rendszerek és Számítógép Architektúrák

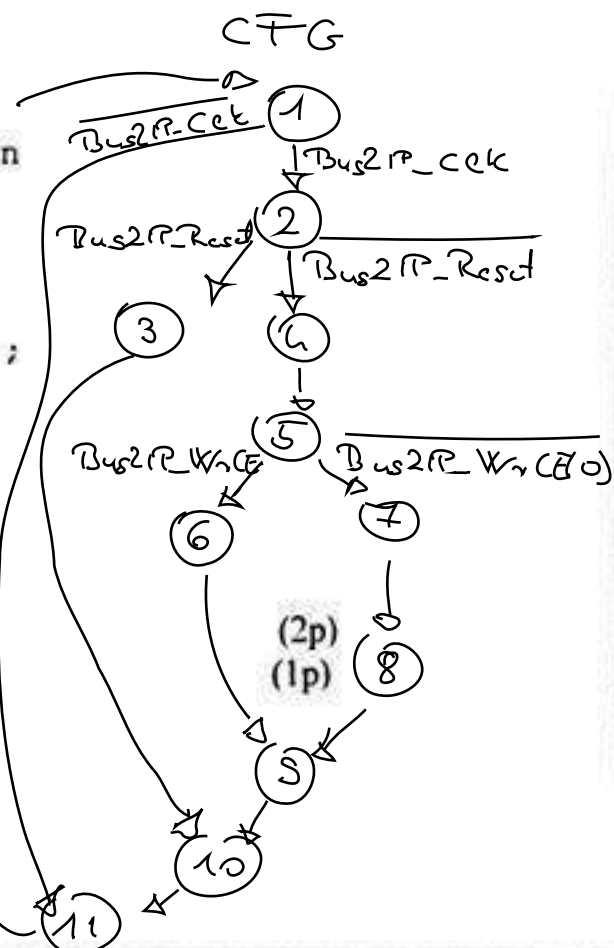
1. ZH
2014. április. 2.

B

1. Magyarozza meg a regiszter ablakozás működését és alkalmazását (3p)
2. Tekintsük a következő paraméterekkel adott 16 bites 2-es komplementes fixpontos rendszert: $p=5$. Mekkora a legkisebb (pozitív) ábrázolható szám? Mekkora a legnagyobb ábrázolható szám? Mekkora a legnegatívabb szám? Mekkora a Δr ebben a rendszerben? (2p)
3. Tervezze meg és rajzolja fel a 4-bites adatbusz Hamming kódos hibajavító áramkörét! (Milyen dekódert kell használni, hogyan kell a paritásbits csoportokat képezni az adat és kódbitek felírásával.) (3p)
4. Adja meg a lebegőpontos osztó blokkdiagramját, az egyes blokkok funkciójával együtt! (2p)
5. Rajzolja fel a egycímű számítógép blokkdiagramját! Definiálja az egyes blokkok funkcióját is! (2p)
6. Adja meg egy 23×23 bites szorzó rendszer blokkdiagramját (7-3) és (3-2) sorcsökkentők egységek felhasználásával, ahol az utolsó szint egy FA. Adja meg a szorzás időszükségletét, ha egy kapu késleltetése G. (3p)
7. Adja meg egy 16 bites LACA (Look-ahead Carry Adder) összeadó blokkdiagramját, és adja meg a 16 bites összeadás időszükségletét (4 bites LACG generátort feltételezve). (3p)
8. Definiálja az allokáció fogalmát (allocation) magas-szintű szintézis esetén! (1p)
9. Adott : memória hozzáférés ideje 20ns, regiszterből regiszterbe másolás 4 ns, kivonás 6 ns, Adja meg a ADD3 X, Y, Z RTL leírását és időszükségletét! (3p)

10. Adja meg az alábbi VHDL szubrutin DFG (adatfolyam) gráfját:

```
LED_PROC : process (Bus2IP_Clk) is
begin
  1 if Bus2IP_Clk'event and Bus2IP_Clk='1' then
  2   if Bus2IP_Reset='1' then
  3     LED_i <= "0000";
  4   else
  5     if Bus2IP_WrCE(0)='1' then
  6       LED_i <= Bus2IP_Data(0 to 3);
  7     else
  8       LED_i <= LED_i;
  9     end if;
 10   end if;
 11 end if;
end process LED_PROC;
```



11. Sorolja fel a kombinációs hálózatok leírási módjait.

1) Magyarázza meg a regiszter ablakozás működését és alkalmazását!

Paraméter átadás "ablakozással": a globális regisztereken keresztül történik, amelyet mindkét (A, B) szubrutin elérhet.

5 bit $\rightarrow$ 32 regiszter
 A(R0 - R31) címezhető meg.
 B(R0 - R31)

közös (globális) regiszterek: R0 - R9
 minden szubrutin által elérhető.

rutin specifikus regiszterek: R10 - R31
 további 3 részből áll
 a regiszterek között történhet átlapolódás.

- Alacsony szintű regiszterek: R10 - R15 (LR)
- Lokális regiszterek: R16 - R25 (LocR)
- Magas szintű regiszterek: R26 - R31 (HR)

A2 eljárás mindaddig jól működik, ameddig a paraméterek száma kisebb a regiszterek mévéte-
 nél, mivel nem igényel memória intenzív Stack műveletet

Fizikai reg. cím

R44
R106
R105
R96
R95
R90
R89
R80
R79
R74

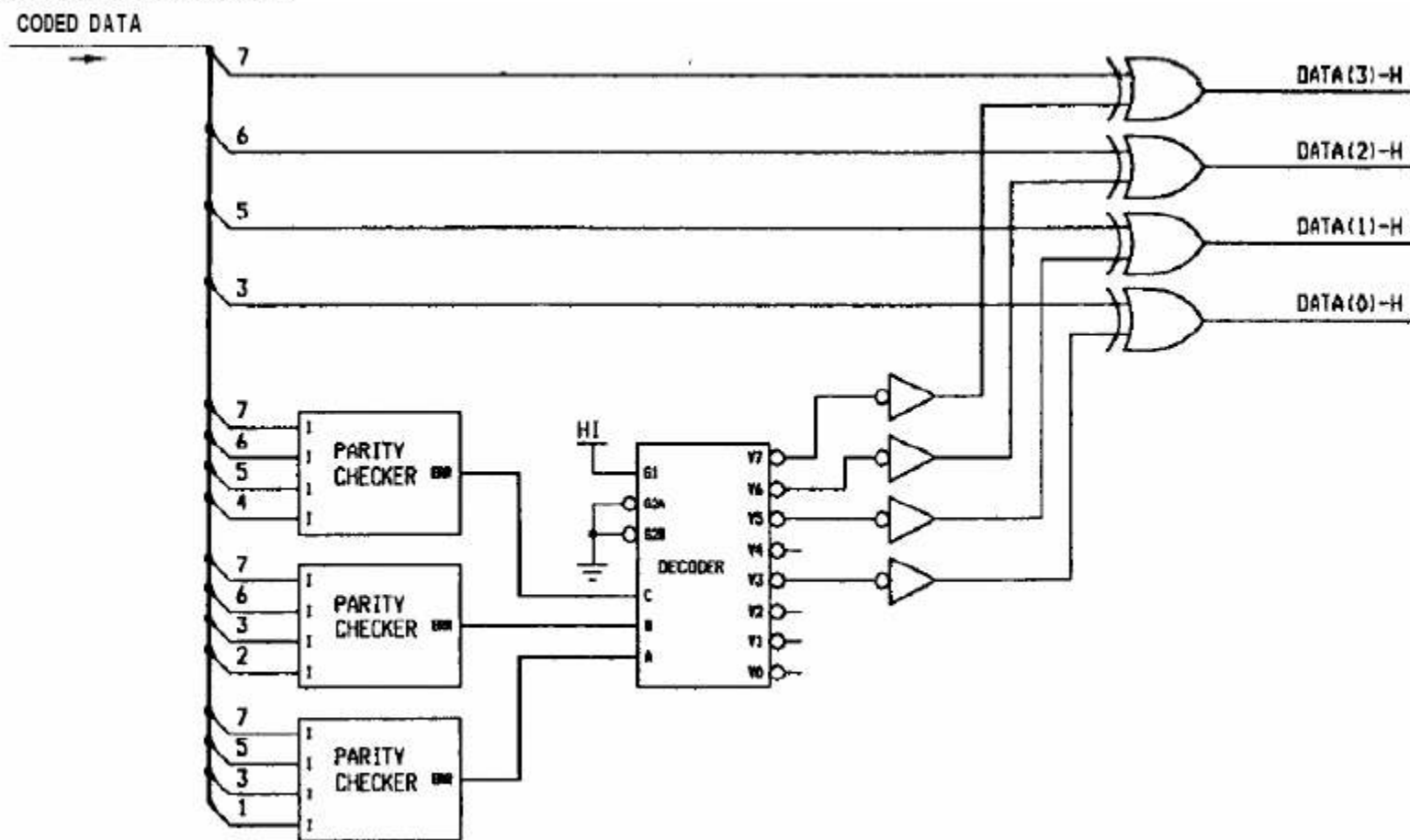
A
R31 R26 R25
HR
LocR
R16 R15 R10
LR

B
R31 R26 R25
HR
LocR
R16 R15 R10
LR

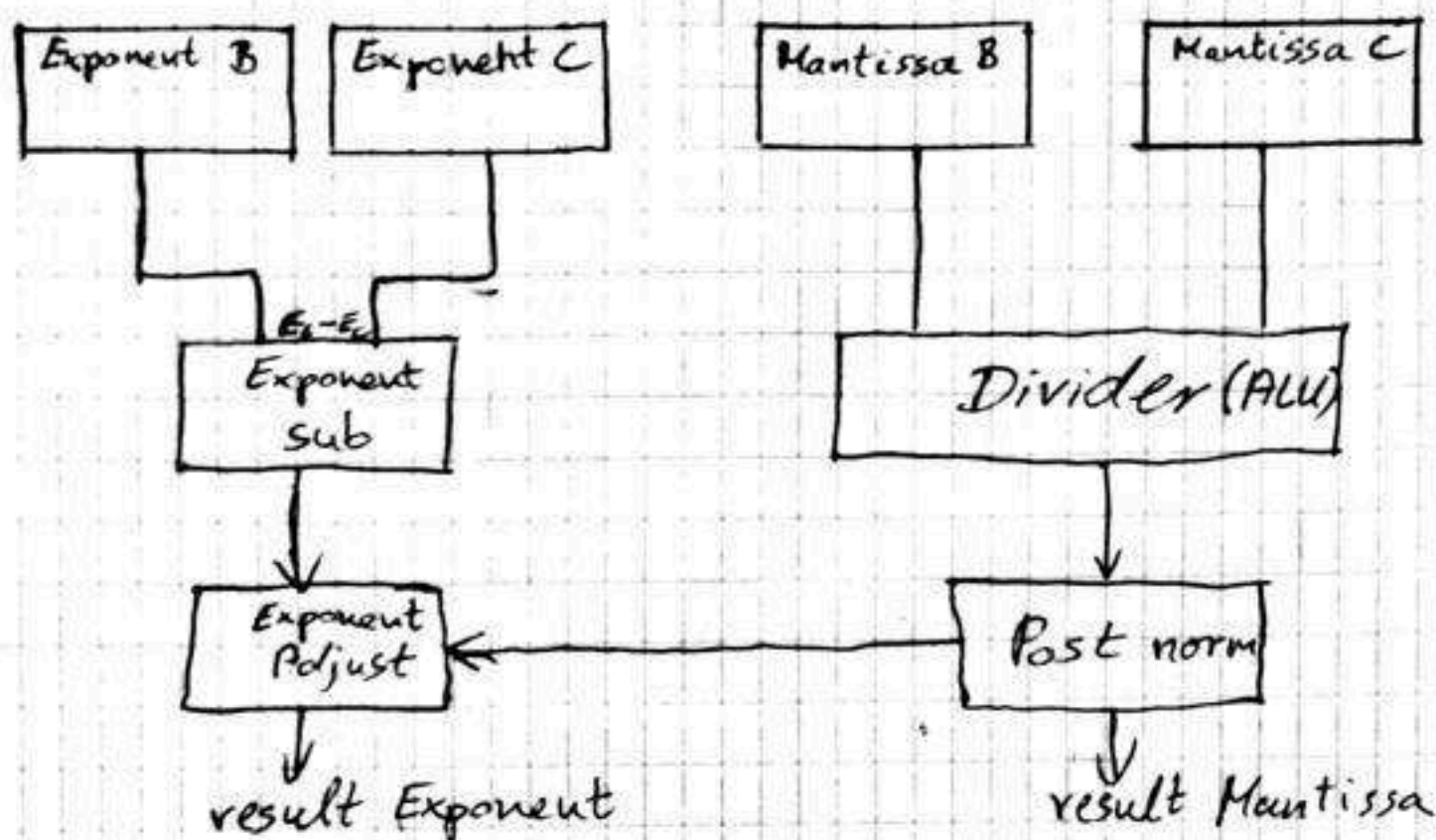
(Fizikailag
 A, B regiszterei
 máshol helyezkednek
 el)

R9
GLOBAL REGISTER
R0

3. feladat Hamming kód



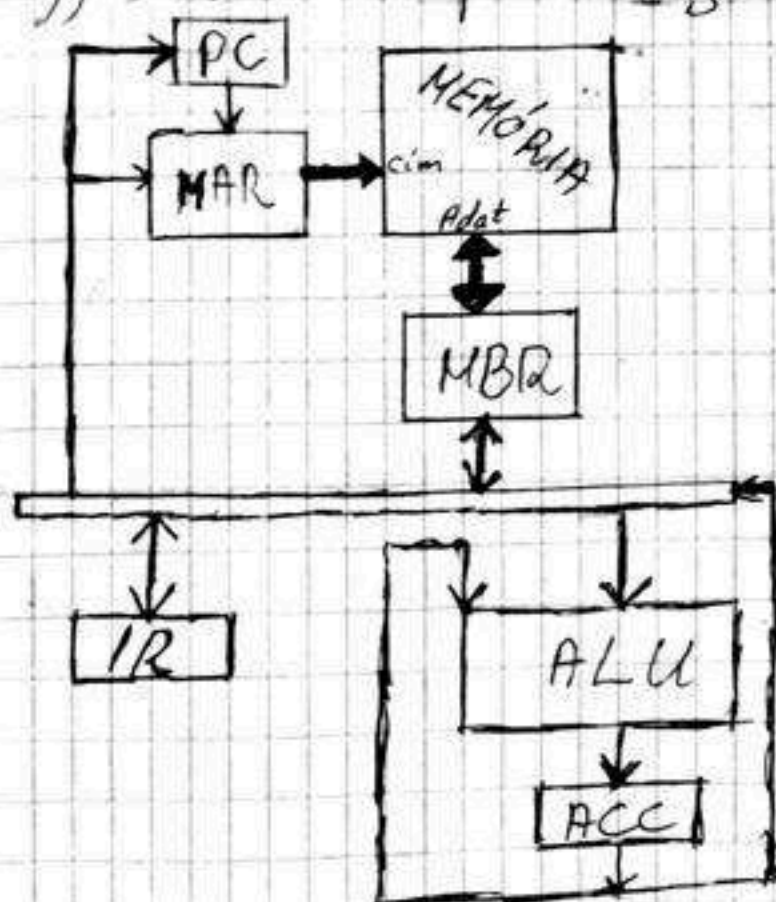
4) Rajolja meg a lebegőpontos osztó blokkdiagrammját, az egyes blokkok funkciójával együtt.



B: osztandó
C: osztó
A: hányados

$$A = B / C$$

5) Rajzolja fel az egycímű gép blokkdiagrammját! Definálja az egyes blokkok funkcióját is.



⑧ Definálja az allokáció fogalmát (allocation) magas szintű szintézis esetén.

- Valamely erőforrás szabad kapacitásának használatba vétele.

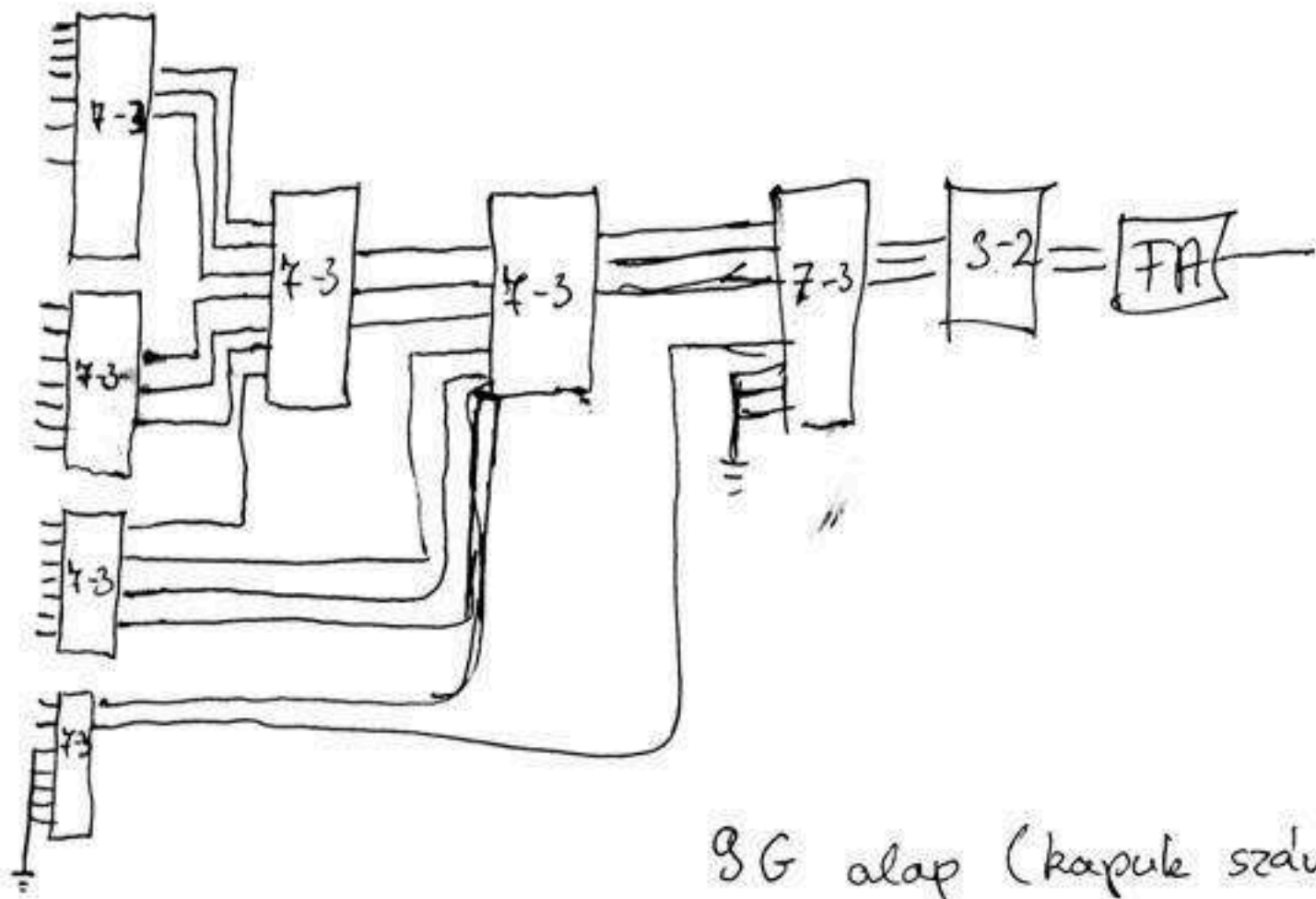
Feladatai: erőforrások feladatokhoz rendelése
erőforrás típusok meghatározása

erőforrás : - műveletvégző
- adattároló
- adatút (MUX, BUSZ)

⑪ Sorolja fel a kombinációs hálózatok leírási módjait.

- Algebrai alak
- Igazságtábla
- algebrai normál alakok
- Karnaugh tábla
- (Quine - McCluskey Módszer ~ igény szerint)

- ⑥ Adja meg egy 23×23 bites szorzó rendszer blokkdiagramját (7-3) és (3-2) sorcsökkentő egységek felhasználásával, ahol az utolsó szint egy fa (FA).
Adja meg a szorzás időszükségletét, ha egy kapu késleltése G .



$3G$ alap (kapuk száma)
 $1G$ (input)

$$3G + 1G + T_{LACH}$$

$$T_{LACH} = 2 + 4 \cdot ([\log_8(56)] - 1)$$

Digitális Rendszerek és Számítógép Architektúrák1. ZH
2014. április 2..**C**

1. Adja meg a RISC processzor jellemző tulajdonságait és adjon meg egy (3p)
2. Tekintsük a következő paraméterekkel adott 15 bites 2-es komplementes fixpontos rendszert: $p=7$. Mekkora a legkisebb (pozitív) ábrázolható szám? Mekkora a legnagyobb ábrázolható szám? Mekkora a legnegatívabb szám? Mekkora a Δr ebben a rendszerben? (2p)
3. Adja meg egy 7 bites sorosan írható párhuzamosan olvasható siftregiszter kapcsolását D tárolók felhasználásával. (3p)
4. Adja meg a lebegőpontos kivonó blokkdiagramját, az egyes blokkok funkciójával együtt! (2p)
5. Rajzolja fel a kétcímű számítógép blokkdiagramját! Definiálja az egyes blokkok funkcióját is! (2p)
6. Adja meg egy 16×16 bites iteratív szorzóalgorithmus folyamatábráját, és ennek áramkörü megvalósítását, továbbá az algoritmus végrehajtási idejét! (3p)
7. Adja meg egy egybites Full Subtractor (FS) igazságtáblázatát, Karnough tábláit, kapusintű kapcsolási rajzát, továbbá a 6-bites RCA (Ripple Carry Adder) kapcsolási rajzát és számítási időszükségletét! (3p)
8. Adja meg a magas-szintű szintézis (HLS) elméleti alaptételét! (1p)
9. Adott : memória hozzáférés ideje 20ns, regiszterből regiszterbe másolás 4 ns, kivonás 6 ns, Adja meg a $ADD2 *X, *Y$ RTL leírását és időszükségletét! (3p)
10. Adja meg az alábbi VHDL szubrutin CFG (vezérlésfolyam) gráfját:

```
TemplAddr : process (clk, state)
begin
    if state = Waiting then
        TemplAddrReg <= 0;
    else
        if clk'event and clk = '1' then
            if ce = '1' then
                TemplAddrReg <= TemplAddrRegNext;
            else
                TemplAddrReg <= TemplAddrReg;
            end if;
        end if;
    end if;
end process;
```

(2p)
- 11 Adja meg a szekvenciális hálózatok leírási módjait.....(1p)

1.) Adja meg a RISC processzor jellemző tulajdonságait
(és adjon meg egy ...?) $\rightarrow$ 16-os tétel 65. old.

Csikentett utasítás kezelési számítógép.

- Csak a kívánt alkalmazásra jellemző utasítás típusokat tartalmaz, ezáltal nő a sebesség
- Minimális utasításkezelési és címzési utasítás, gyors HW elemeket, optimalizált SW kóddal.
- Szubrutinok és hosszabb utasításokhoz van szükség (sok egyszerű utasítás, a komplex feladatok miatt)
- Azonos hosszúságú utasításformátum
- Hatalmas utasítás dekódolás
- Egyszerű ábrák megtekintése - VLSI technológiához
- LOAD/STORE memóriaaccess: Látványos - töltés és tárolás
A regiszterek gyorsabbak, mint a memória-intenzív műveletek
- pipeline (párhuzamos utasításfeldolgozás), többszörös adatvonalak, nagyszámú gyors regiszter alkalmazásával.

2.) Tekintve a körtekész paraméterekkel adott 15 bites
2-es komplementű fix pontos számot, $p=7$.

a.) Mekkora a legkisebb pozitív ábrázolható szám?

00000000,0000001

b.) Mekkora a legnagyobb ábrázolható szám?

01111111,1111111

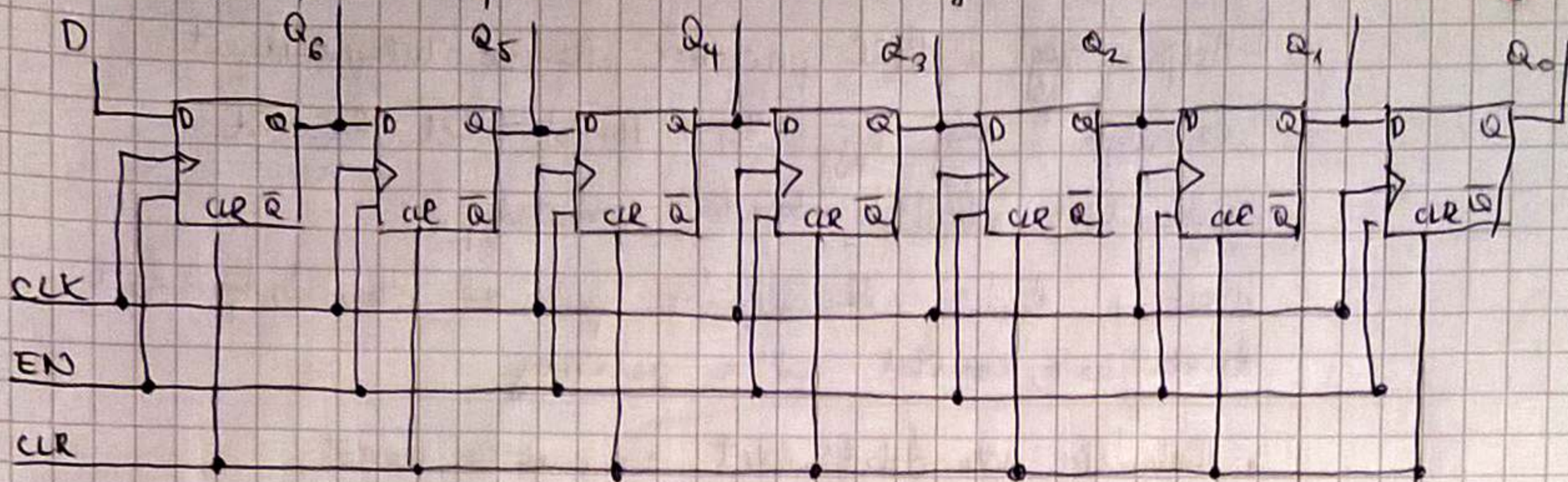
c.) Mekkora a legnegatívabb szám?

11111111,1111111

d.) Mekkora Δr a rendszerben?

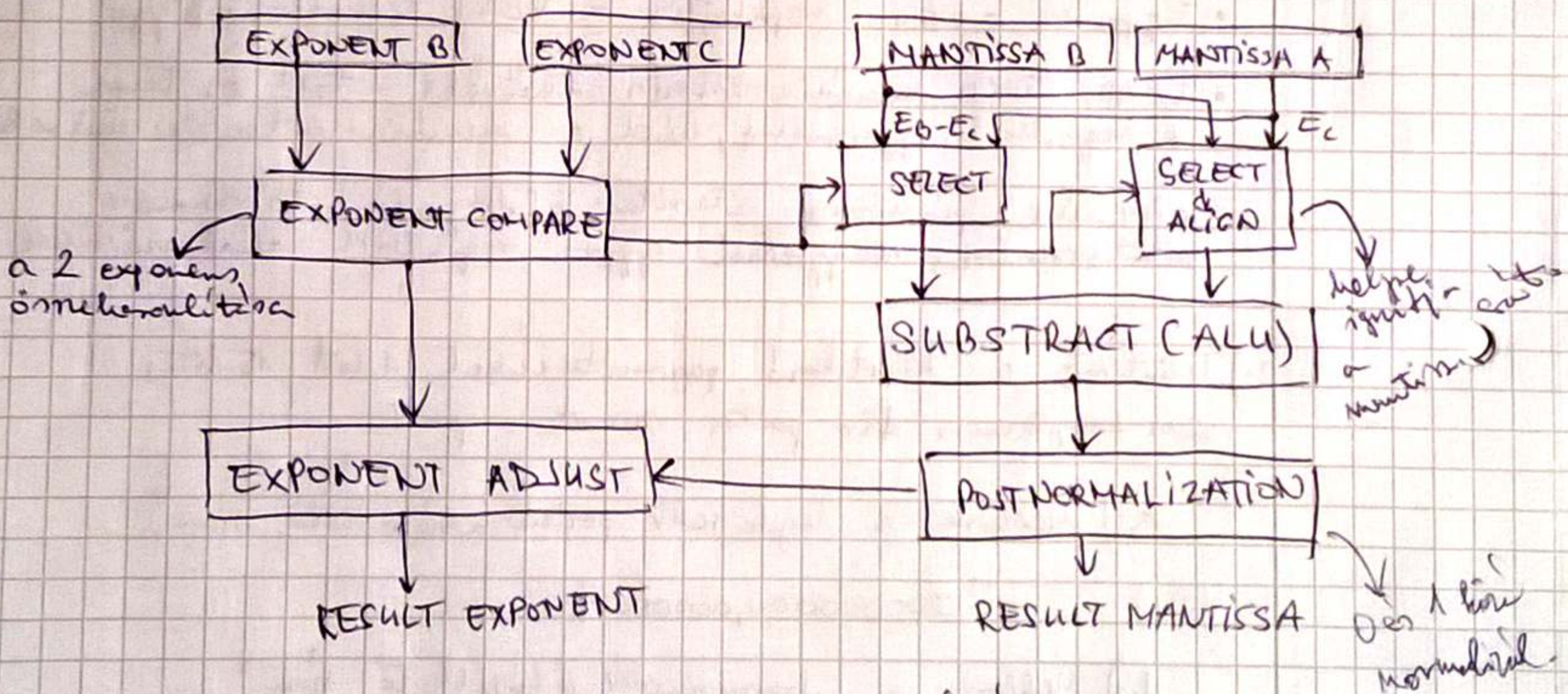
$$\Delta r = 2^{-p} = 2^{-7} = \frac{1}{128}$$

3.) Adja meg egy 7 bites sorosan írató/párkuszorosan olvasható shiftregiszter kapcsolását D-tárolók felhasználásával.



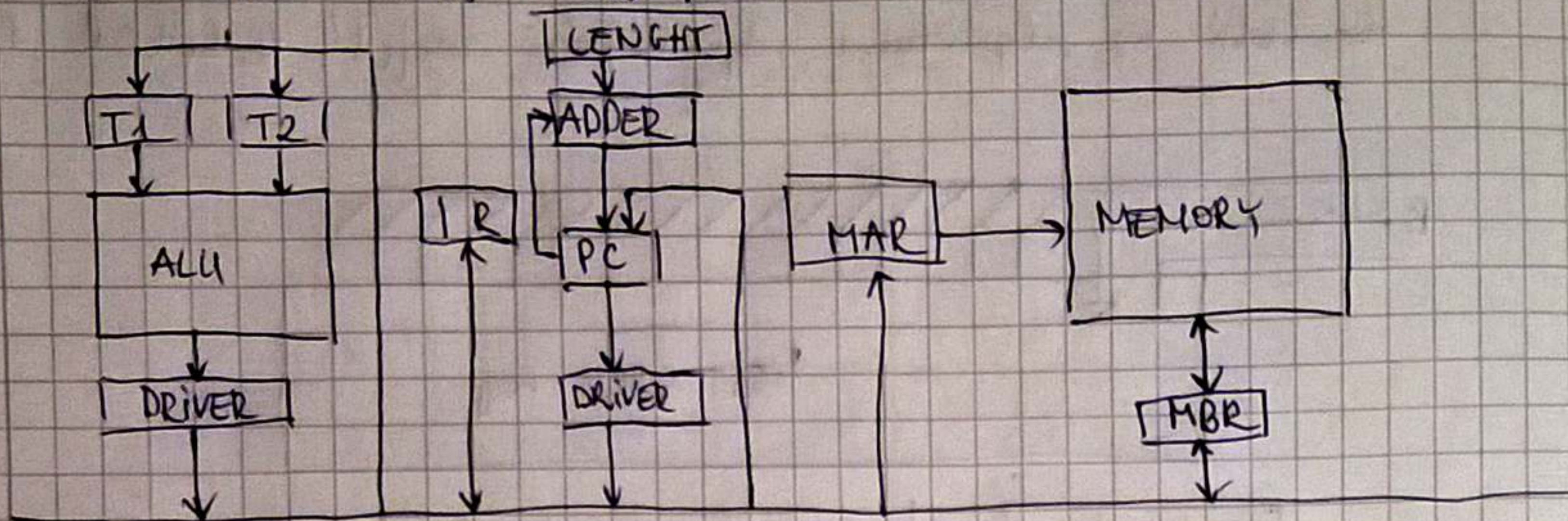
4.) Adja meg a lebegőpontos kivonás blokkdiagramját, az egyes funkciókhoz együtt (0 < B < C)

$$A = B + C = M_B \times r^{E_B} - M_C \times r^{E_C} = (M_B \times r^{E_B - E_C} - M_C) \times r^{E_C}$$



A blokkdiagramm aritmetikai ~~egység~~ a **SUBTRACT (ALU)** egység jelenti.

5.) Rajzolja fel a letező műveletgép blokkdiagramját!
 Definíálja az egyes egységeket! 8. tétel 29. old.



Memória - (tároló nével lineáris tömbje). A memória nével
 általában olyan nével, amelyen nével az adatok.
 Pl. m -bit nével egy nével és álljon M bit névelből.
 Ekkor: $\log_2(M)$ nével címveztéssel elérhető meg
 a memória.
 Az adatok nével (írás/olvasás) névelitáció
 is megengedett.

A memória Neman architektúrát követi:

- az utasítások (program) és az adatok egy nével
 tárolóba, nem pedig külön-külön (Harvard architektúra).
 A programot is adatként tárolja.

MAR - (Memória cím regiszter): a memória nével érkező vagy
 a memória nével lévő info névelit áronótiá memória nével
 alapján.

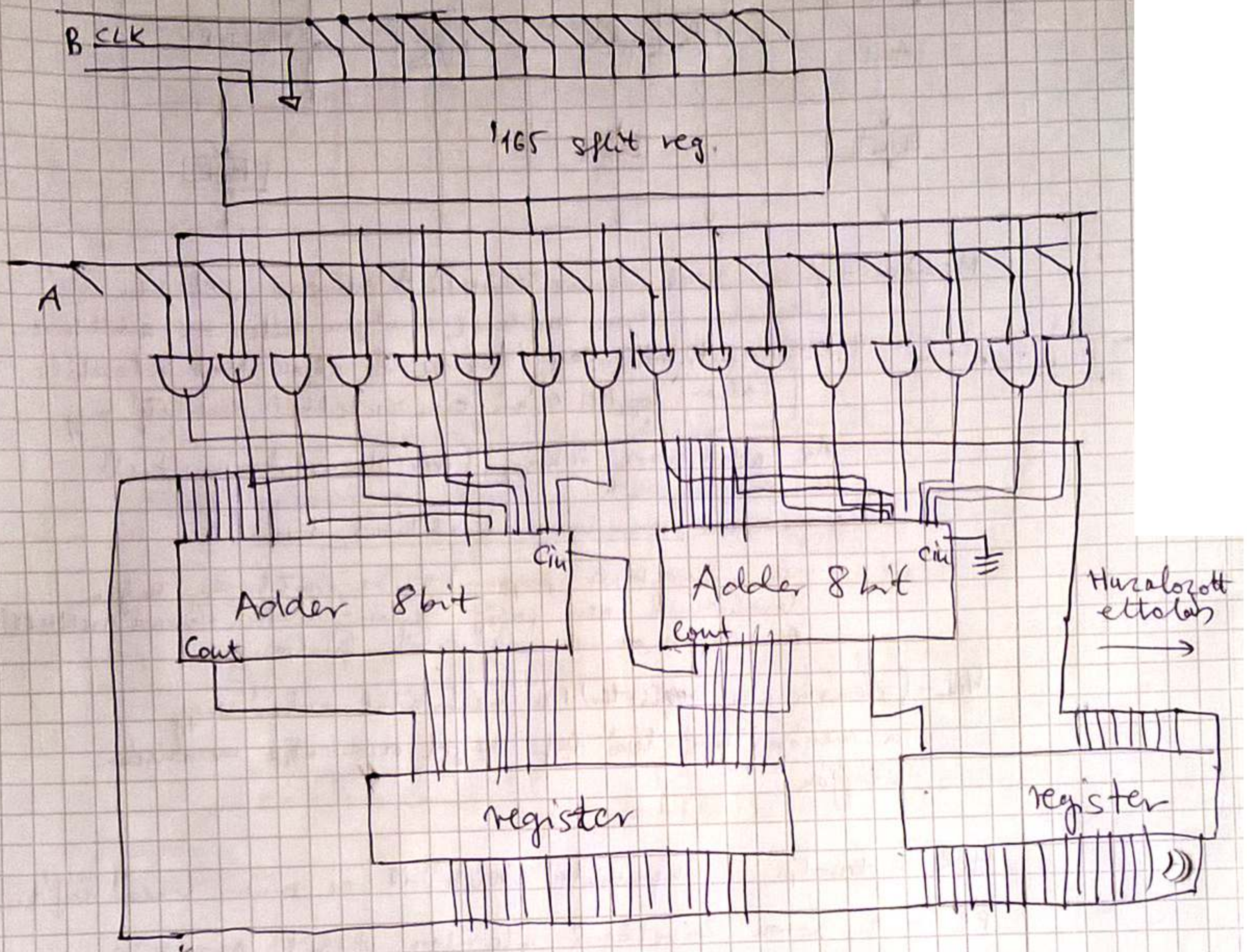
MBR - tárolja a memória nével bevitt, ill. az a nével érkező info.

PC - a soron névelkövetkező utasítás névelit áronótiá.

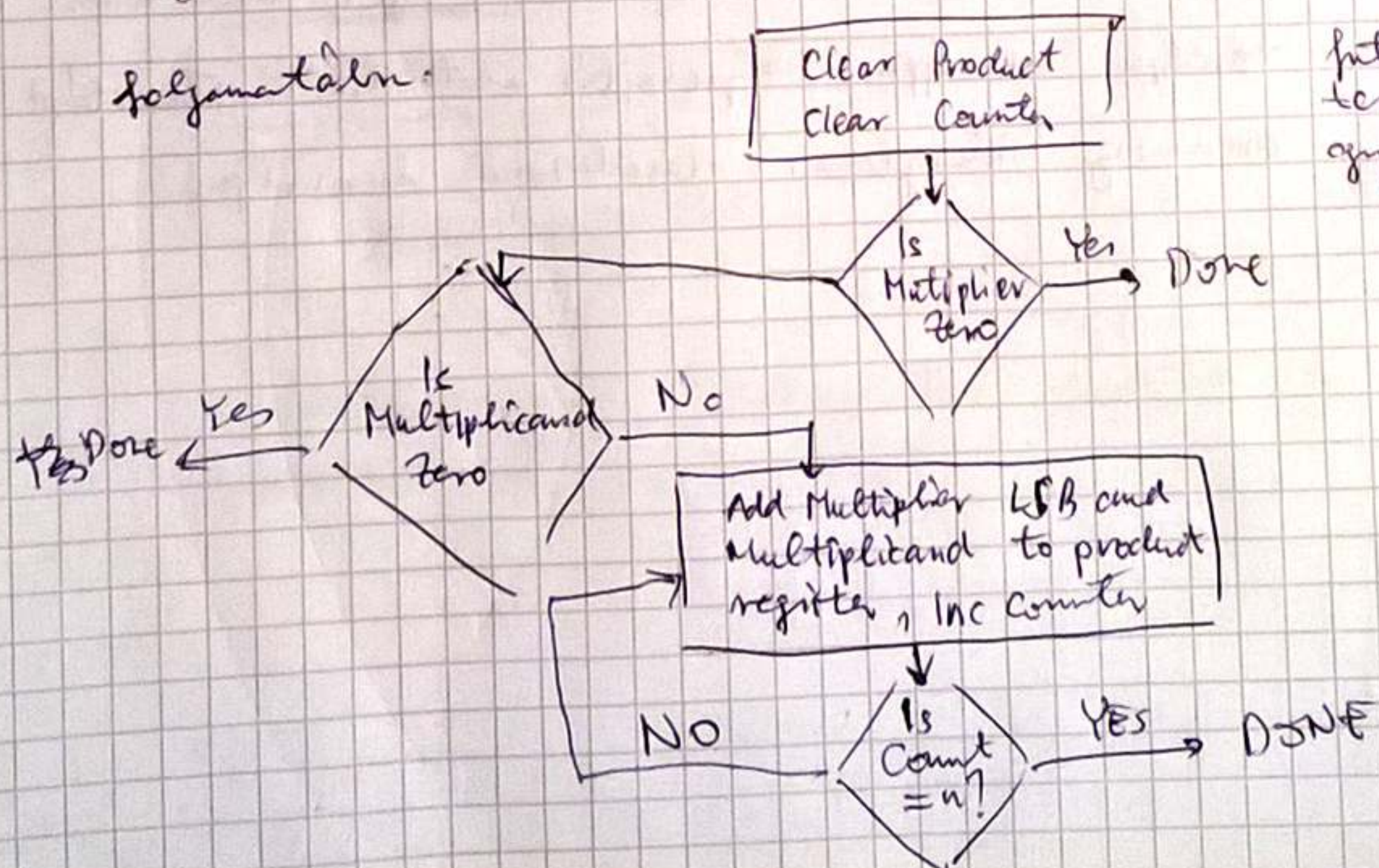
IR - tárolja az éppen névelhajtás alatt lévő utasítást.

ACC - eredmény ideiglenes tárolására névelit.

6.) Adja meg egy 16×16 bites iteratív szorzóalgoritmus folyamatábráját és ennek áramkörti megvalósítását, továbbá az algoritmus végrehajtási idejét! 4. tétel 17. old.



szorzatán.



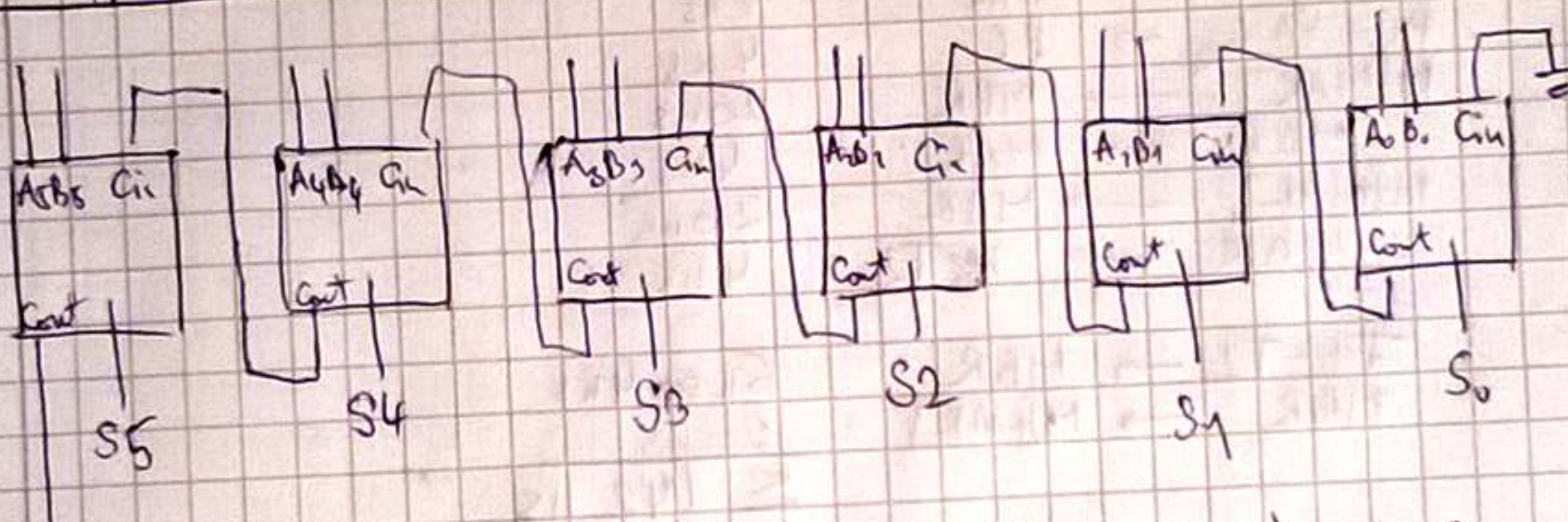
fültési ideje
tartalékai feltételekkel,
egyszerűsítések
Pl. ha 0-val
szorozunk nem megy,
mert az eredmény
is 0 lesz.

7.) Adja meg egy egy bites FS igazságtáblázatát, karrouge táblázatát, kapu szintű kapcsolási rajzát + a 6 bites RCA kapcsolási rajzát és számítási időmértékelést.

	A	B	Bin	F	Bout
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

F:		B	
0	1	0	1
1	0	1	0
2	1	0	1
3	0	1	0
4	1	0	1
5	0	1	0
6	1	0	1
7	0	1	0

A		B	
0	1	0	1
1	0	1	0
2	1	0	1
3	0	1	0
4	1	0	1
5	0	1	0
6	1	0	1
7	0	1	0



$$\text{Time RCA} = N \cdot T_{FS} = N \cdot (2 \cdot t) = 6 \cdot (2 \cdot t) = 12t$$

8.) Adja meg a magas szintű szintézis (HLS) ^{elméleti} vázlatát 18. tétel. 68. old.

~~High level synthesis (HLS) is a technique for converting a high-level description of a system (e.g., C or C++) into a low-level hardware description (e.g., Verilog or VHDL). It is used to simplify the design of digital systems.~~

~~High level synthesis (HLS) is a technique for converting a high-level description of a system (e.g., C or C++) into a low-level hardware description (e.g., Verilog or VHDL). It is used to simplify the design of digital systems.~~

Turing Church tétel: A szimbolikus számítást 3 fő elemre bontva lehet leírni: a következők:

- Turing gép (TM - hardware)
- μ -reprezentáció (algoritmusok, programok)
- CFG - környezet független (DS)

Egy nyelv, mely leírja a teljes rendszer működését
az egy rendszer leírását jelenti: NYELV $\Leftrightarrow$ HW

9.) Adott: memória hozzáférések ideje 20ns, reg-lől reg-ka másolás 4ns
 kivétel 6ns. Adja meg a $ADD2\ *X, *Y$ RTL leírásait
 és időmikrosfelteket. Fittel 33. old.

Fetch:

PC $\rightarrow$ MAR	4ns
M[MAR] $\rightarrow$ MBR	20ns
PC+length $\rightarrow$ PC	4ns
MBR $\rightarrow$ IR	4ns

Decode: felkutatni k, hogy 0

Execute:

PC $\rightarrow$ MAR	4ns
PC+XA _{6n} $\rightarrow$ PC	4ns
M[MAR] $\rightarrow$ MBR	20ns
MBR $\rightarrow$ MAR	4ns
M[MAR] $\rightarrow$ MBR	20ns
MBR $\rightarrow$ T1	4ns

PC $\rightarrow$ MAR	4ns
PC+YA _{6n} $\rightarrow$ PC	4ns
M[MAR] $\rightarrow$ MBR	20ns
MBR $\rightarrow$ MAR	4ns
M[MAR] $\rightarrow$ MBR	20ns
MBR $\rightarrow$ T2	4ns

T1-T2 $\rightarrow$ MBR	6ns+4ns
MBR $\rightarrow$ M[MAR]	20ns
Σ	142 ns

10.) Adja meg az alábbi VHDL szintetizált CFG vezérlő logika
 rajzát.

