Digszar 1.ZH 2018 A

1. Adja meg a RISC processzor jellemző tulajdonságait és adjon meg egy példát.

   - Reduced Instruction Set Computer (Csökkentett utasításkészletű számítógép)
   - csak a kívánt alkalmazásra jellemző utasítástípusokat tartalmaz
   - minimális utasításkészlet és címzési mód, gyors hardware, optimalizált software
   - szubrutinok és hosszabb utasítássorozatok
   - azonos hosszúságú utasításformátum
   - huzalozott utasításdekódolás miatt kombinációs logika
   - egyszeres ciklusvégrehajtás: 1 utasítás / 1 ciklus
   - LOAD/STORE memóriaszervezés: töröl és tárol
   - pipeline technika – utasításfeldolgozás párhuzamosítása

   PÉLDA: Motorola 88000, Alpha, SPARC, PIC, DSPR sorozatok, IBM PowerPC stb.

2. Tekintsük a következő paraméterekkel adott 15 bites 2-es komplemens fixpontos rendszert: p=7.
   a. Mekkora a legkisebb pozitív ábrázolható szám?

   $$0000000{,}0000001_B = 2^{-7} = 0.0078125_D$$

   b. Mekkora a legnagyobb ábrázolható szám?

   $$01111111{,}1111111_B = 2^6+2^5+2^4+2^3+2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}+2^{-7} = 127{,}99 \approx 128_D$$

   c. Mekkora a legnegatívabb szám?

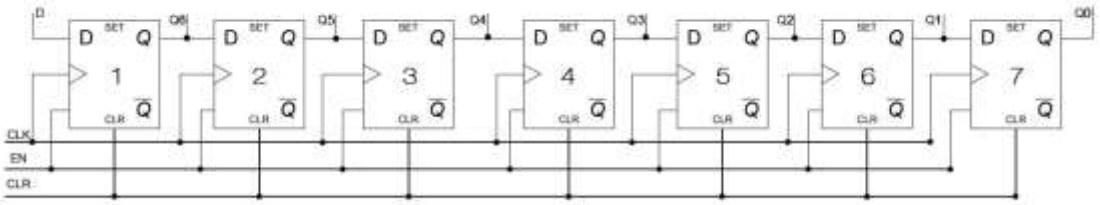
   $$10000000{,}0000000_B = -128_D$$

   d. Mekkora az Δr (differencia)?

   $$\Delta r = 2^{-P} = 2^{-7} = 0.0078125$$

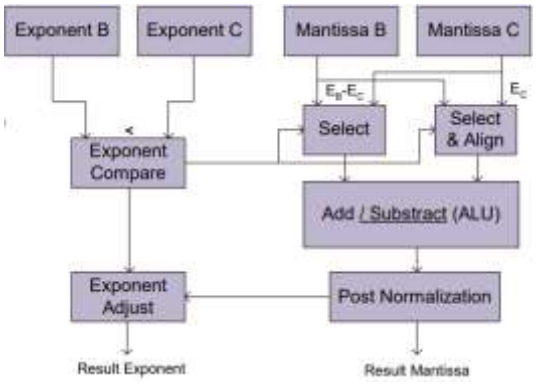
3. Adja meg egy 7 bites sorosan írható párhuzamosan olvasható shiftregiszter kapcsolását D tárolók felhasználásával.

   D tároló: egy bemenete van (D), két kimenetel: Q és nem-Q. Szükséges még CLR, CLK és Enable.



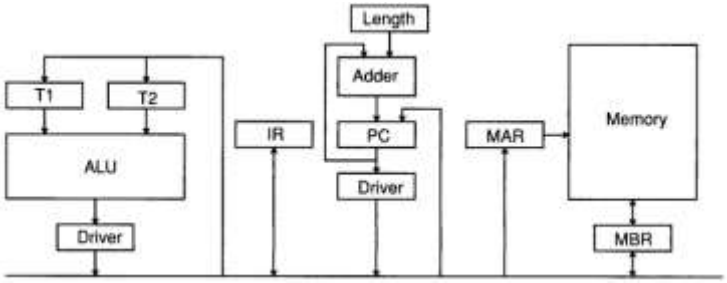
4. Adja meg a lebegőpontos kivonó blokkdiagramját, az egyes blokkok funkcióival együtt!

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



Exponent B, C: B és C exponense, kitevője
Mantissa B, C: B és C alapja, mantissája
Exponent compare: a magasabb kitevőt választja
Select/Select Align: mantissa igazítása
ALU: elvégzi a művelet
Post Normalization és Exponent Adjust: eredmény normalizálása

5. Rajzolja fel a kétcímű számítógép blokkdiagramját! Definiálja az egyes blokkok funkcióját is!

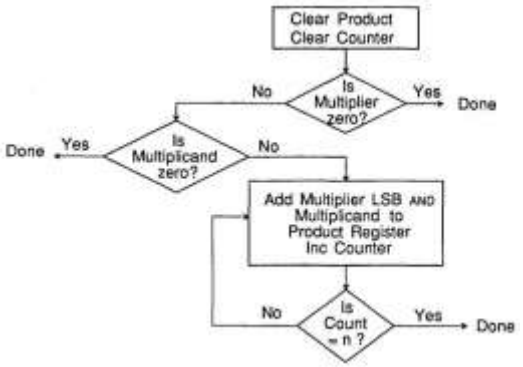


T1, T2: regiszterek
ALU: Aritmetikai Logikai Egység. Matematikai és logikai műveleteket hajt végre
Driver: Meghajtó program
IR: Utasításregiszter. A lehívott utasítást tárolja a dekódolás idejére.
Length: utasítás hossza. Ezzel szükséges megnövelni a PC-t.
PC: Program Counter. Utasítás számláló regiszter. A soron következő utasítás memóriabeli címét tartalmazza.
MAR: Memory Address Register. Memóriába/ból áramló információ címe.
MBR: Memory Buffer Register. Memóriába/ból áramló információ tárolása.

A regiszteres kétcímes számítógépgép kiegészül: Regiszter Bank és Regiszter Select.

6. Adja meg a 16*16 bites iteratív szorzóalgoritmus folyamatábráját és ennek áramköri megvalósítását, továbbá az algoritmus végrehajtási idejét!
   a. folyamatábra

   A Shift and Add egy hagyományos iteratív szorzási módszer, hasonlít a decimális papíron történő szorzáshoz. Gondolom ide ez egy elfogadott példa lehet, így ennek a folyamatábrája itt van:

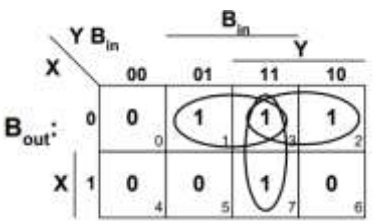


   b. áramköri megvalósítás – még rajzolom
   c. Algoritmus végrehajtási ideje:

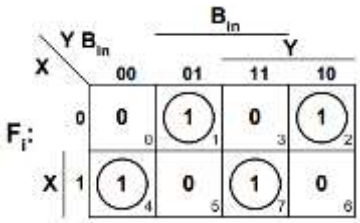
Tmult = Tsetup + N * Titer, ahol Titer = Tand + Tsum + Treg

7. Adja meg egy egybites Full Substractor (FS) igazságtáblázatát, Karnough tábláit, kapuszintű kapcsolási rajzát, továbbá a 6-bites RCA (Riple Carry Adder) kapcsolási rajzát és számítási időszükségletét!

Xi	Yi	Bin	Fi	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



$$B_{out} = \overline{X_i} \cdot Y_i + \overline{X_i} \cdot B_{in} + Y_i \cdot B_{in}$$

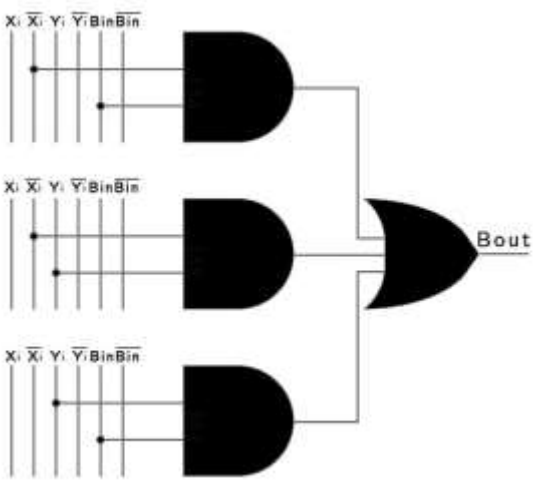


$$F_i = X_i \oplus Y_i \oplus B_{in}$$

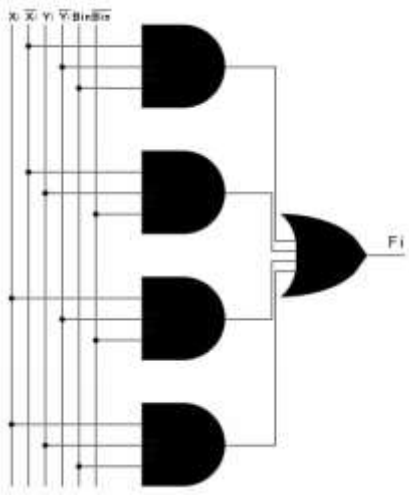
Fi = Xi XOR Yi XOR Bin felírható úgy is, hogy:

Fi = Xi'Yi'Bin + Xi'YiBin' + XiYi'Bin' + XiYiBin (Yi' = Yi negált)

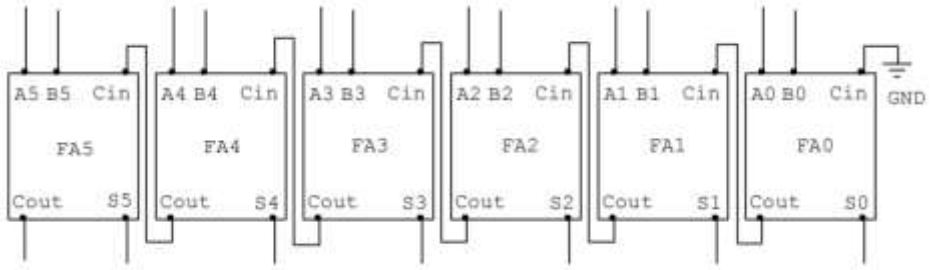
Bout kapcsolási rajz:



Fi kapcsolási rajz:



6 bites RCA rajza és számítási időszükséglete:



$T_{(RCA)} = N * T_{(FA)} = N * (2G)$. Ha RCA 6 bites, akkor N = 6, azaz

$$T_{(RCA)} = 12\ G$$

8. Adja meg a magas-szintű szintézis (HLS) elméleti alaptételét!

   a. Turing gép (Turing-Machine, hardver)
   b. μ-rekurzív függvények (algoritmus)
   c. környezetfüggetlen nyelvtan (programozási nyelvek)

   A Turing-Church tézis kimondja, hogy minden formalizálható probléma, amely megoldható valamilyen algoritmussal megoldható Turing géppel vagy valamilyen vele ekvivalens absztrakt modellel pl: környezet független nyelvtannal is. Tehát létezik olyan transzformáció, amely egy magas szintű nyelven leírt algoritmust áramköri leírássá alakít át.

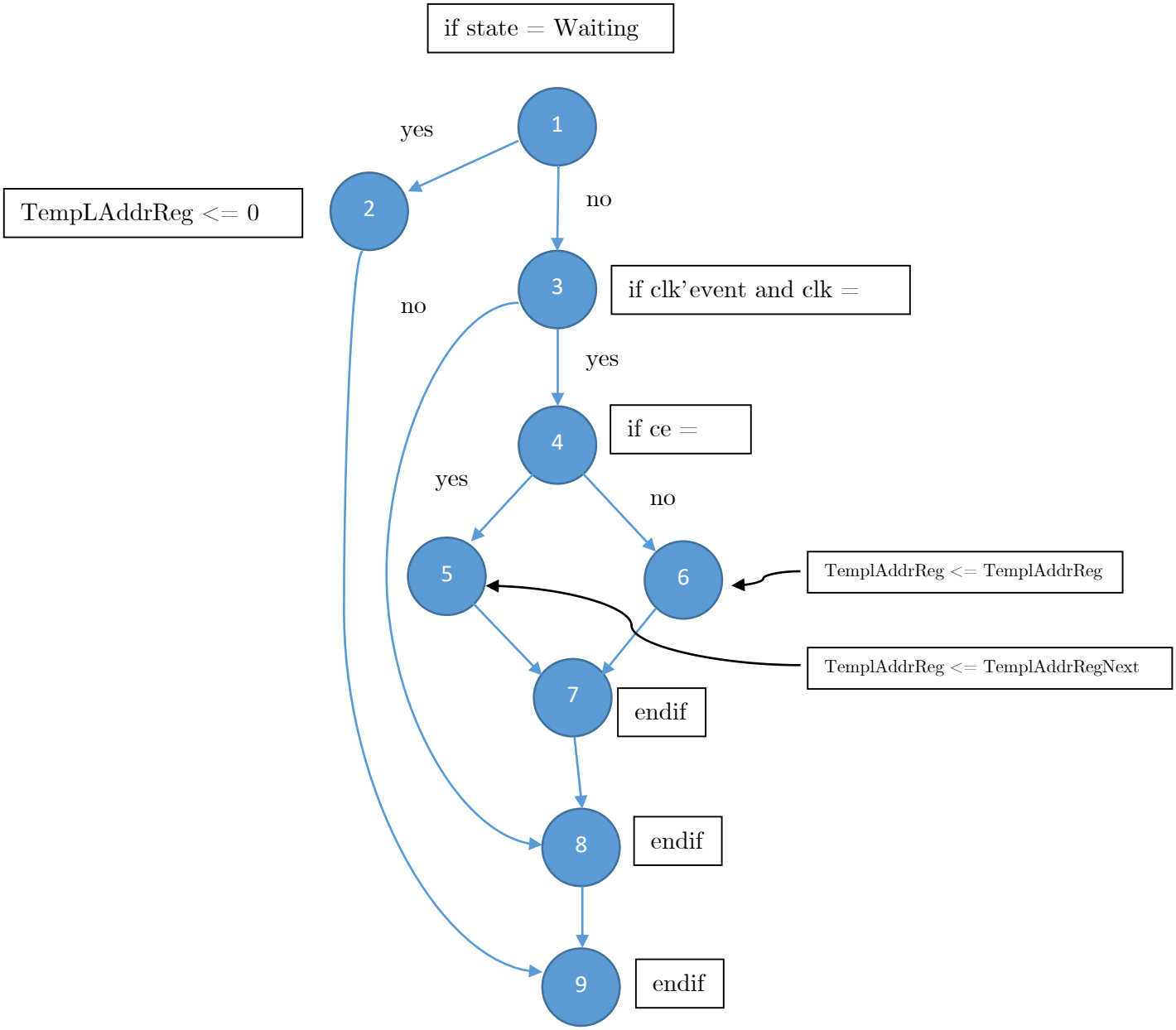
   LINK

9. Adott: memória hozzáférés ideje 20ns, regiszterből regiszterbe másolás 4ns, kivonás 6s. Adja meg a ADD2 *X, *Y RTL leírását és időszükségletét! (Decode idejét 0-nak tekintjük.)

Fetch	Execute	
4ns: PC → MAR	4ns: PC → MAR	20ns: M[MAR] → MBR
20ns: M[MAR] → MBR	4ns: PC + XA_len → PC	4ns: MBR → MAR
4ns: PC + I_len → PC	20ns: M[MAR] → MBR	20ns: M[MAR] → MBR
4ns: MBR → IR	4ns: MBR → MAR	4ns: MBR → MAR
12 + 20 = 32ns	20ns: M[MAR] → MBR	20ns: M[MAR] → MBR
	4ns: MBR → MAR	4ns: MBR → T2
	20ns: M[MAR] → MBR	6ns: T1 – T2 4ns: → MBR
	4ns: MBR → T1	20ns: MBR → M[MAR]
	4ns: PC → MAR	44 + 140 + 6 = 190ns
	4ns: PC + YA_len → PC	32ns + 190ns = 222ns

10. Adja meg az alábbi VHDL szubrutin CFG (vezérlésfolyam) gráfját:

```
TemlAddr : 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;
```



11. Adja meg a szekvenciális hálózatok leírási módjait.

Igazságtábla, algebrai kifejezések, elvi logikai vázlat, állapottábla, állapotgráf stb.