

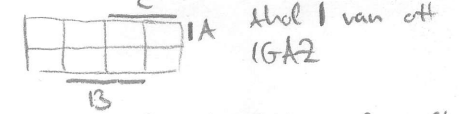
Bináris számrendszer  
 -> 2-es számrendszer, sorozás  
 Lettes számrendszer: 1) invariáns  
 2) +1  
 -> könnyű aritmetikai tesztek

Boole algebra  
 -> ÉS  $x \Rightarrow D$   
 -> VAGY  $+ \Rightarrow D$   
 -> NEM  $- \Rightarrow D$   
 -> XAND  $\Rightarrow D$   
 -> XOR  $\Rightarrow D$

Egységelem:  $B \cdot 1 = B$   $B + 0 = B$   
 Nullaelem:  $B \cdot 0 = 0$   $B + 1 = 1$   
 Idempotencia:  $B \cdot B = B$   $B + B = B$   
 Involúció:  $\overline{\overline{B}} = B$   
 Komplement:  $B \cdot \overline{B} = 0$   $B + \overline{B} = 1$   
 Kommutativitás:  $B \cdot C = C \cdot B$   $B + C = C + B$   
 Asszociativitás:  $(B \cdot C) \cdot D = B \cdot (C \cdot D)$   
 $(B + C) + D = B + (C + D)$   
 Disztributivitás:  $(B \cdot C) + (B \cdot D) = B \cdot (C + D)$   
 $B + (C \cdot D) = (B + C) \cdot (B + D)$   
 Eltagolás:  $B \cdot (B + C) = B$   $B + (B \cdot C) = B + C$   
 Összevonás:  $(B \cdot C) + (B \cdot \overline{C}) = B$   
 $(B + C) \cdot (B + \overline{C}) = B$   
 De Morgan:  $\overline{B \cdot C} = \overline{B} + \overline{C}$   $\overline{B + C} = \overline{B} \cdot \overline{C}$

De Morgan:  $A + B = \overline{\overline{A} \cdot \overline{B}}$   $\overline{A} \cdot \overline{B} = \overline{A + B}$   
 $\overline{B_0 + B_1 + \dots + B_n} = \overline{B_0} \cdot \overline{B_1} \cdot \dots \cdot \overline{B_n}$   
 $\overline{B_0 \cdot B_1 \cdot \dots \cdot B_n} = \overline{B_0} + \overline{B_1} + \dots + \overline{B_n}$

Algebrai normál alak  
 DNF = SoP: Sum of Products  
 CNF = PoS: Product of Sums

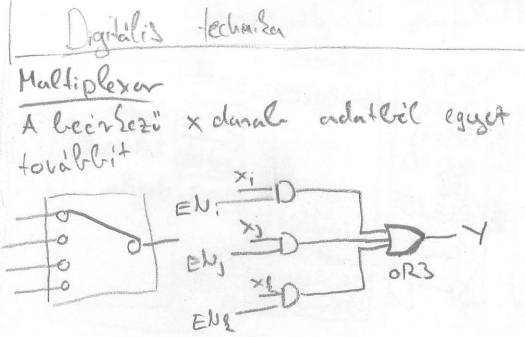
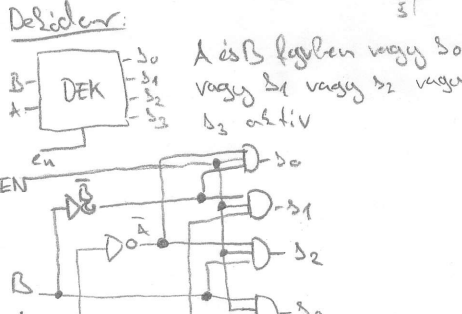


Bináris 2<sup>n</sup> darabot egybe rakhatunk  
 amikor figy minimalizálunk

Többszintű realizáció  
 SoP és PoS többszintű realizáció

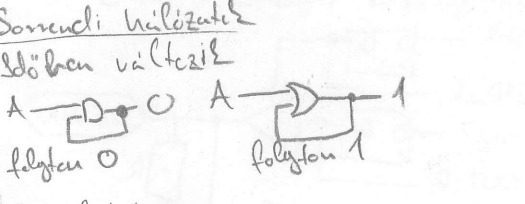
$F = ABC + ABD + \overline{A}BC + \overline{A}BD$   
 -> 4db AND3 és 1db OR4  
 $x = AB$   $y = C + D$   
 $F = xy + \overline{x}\overline{y}$   
 -> 3db AND2 2db OR2

Funkcionális egységek  
 • DEC, Max, (DE)Max  
 • ADD, SUB, COMP



Demultiplexer  
 Egy adatforrást több kimenetre oszt szét

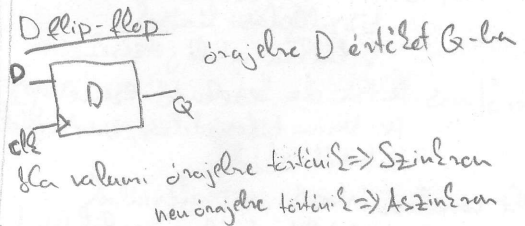
Univerzális logikai elemek  
 Minden elem amivel az összes AND, OR, NOT-tal megvalósítható figy meg lehet valósítani:  
 PR: NAND  
 NOR  
 Mux  
 Memória táblázat LUT



S-R latch

SET-RESET latch

S(A)	R(A)	Q(A)	Q(A+)
0	0	Q(A)	Q(A)
0	1	x	0
1	0	x	1
1	1	x	1



Regiszter

alvagy @ (posedge clk)  
 if (RESET) Q = 1'b0;  
 else if (LOAD) Q = D;

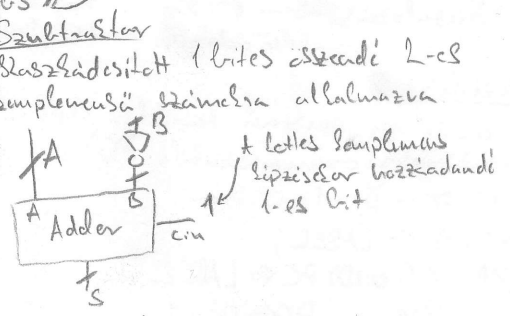
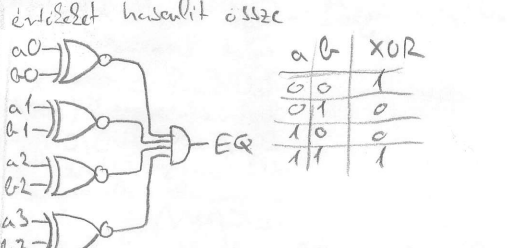
Sorrendi hálózat  
 Állapotok egymásutánján  
 clk-ra state(t) = state(t+1)  
 Mealy modell: A kimenet függhet a bemenettől is  
 Moore modell: A kimenet csak az állapottól függ

Shiftregiszter  
 A regiszter bitjeit jobbra vagy balra shift el.  
 Segít soros/párhuzamos adattárolásban

Szimbóliz  
 fel/lefelé szimbóliz

alvagy @ (posedge clk)  
 begin  
 if (rst)      q = 8'b0;  
 else if (load)      q = d;  
 else if (en)      q = q + 8'b1;  
 if (dir)      q = q - 8'b1;  
 else

end  
 assign tc = en & (dir ? (8q) : (~1q));

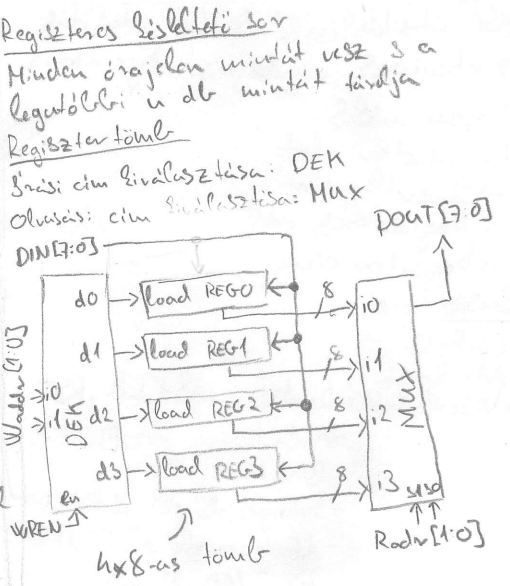


Szabványosított/Dezintermet  
 B=0 cin=1  
 -> S=A+1

Használati táblázat  
 Használati táblázat: dezintermet

Adattárolás  
 Regiszter alapú: regiszteres létezők sor  
 regiszter tárol

Memória: ROM, RAM (Szinkron), RAM (Aszinkron)

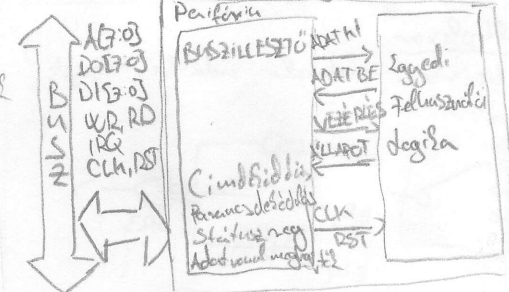


Memória  
 Read Only Memory  
 Random Access Memory  
 Stack: LIFO (Last In First Out)  
 Sor: FIFO (First In First Out)  
 Parancsok: Push  
 Pop

# Digitális rendszer átfutása

- Adatstruktúra + Vezérlés
- Adatstruktúra
  - Adathárrolás: Összeállítások; leírás; leírás; leírás; leírás
  - Regiszter: cím; hozzáférést; cím; cím; cím
  - Stack: cím; a stack; felje érhető el
- Művelet végző:
  - Száj: elem; leírás; elem; leírás
  - Töltés: cím; cím; cím; cím
  - Átírási: cím; cím; cím; cím
  - Átírási: cím; cím; cím; cím
- Átírási: cím; cím; cím; cím

# Átírási tulajdonságok



Átírási cím: Saját pontos címjelre  
 kiadja a bitet / mintát  
 a vonalat  
 Mintavétel: a STOP bit értéket is

## Vezérlés

A vezérlő egy egyszerű számítógép alapú  
 vezérlőegység (programszámláló, PC)  
 CONT:  $PC \leftarrow PC + 1;$   
 JUMP:  $PC \leftarrow LABEL;$   
 JMP: if (COND)  $PC \leftarrow LABEL$   
 else  $PC \leftarrow PC + 1$

PC új értéke → FETCH → DECODE → EXECUTE

## Processzor utasítások

- Adatmozgató: MOV, vagy LOAD/STORE
- Aritmetikai: AND, SUB, COMP, MUL, DIV
- Logikai: AND, OR, XOR, CPL, TST
- Leíró: SLX/SRX, RR/RL, ASR és aritmetikai
- Vezérlő: JMP, CJP, JAL, CALL, RTN
- Száj: NOP, EI, DI, HALT

RISC utasítások: 50-150 utasítás  
 Az utasítások végrehajtási ideje azonos

## Címzési módok

- Szóvalen adat
- Szóvalen cím
- Regiszteres adat
- Regiszteres cím

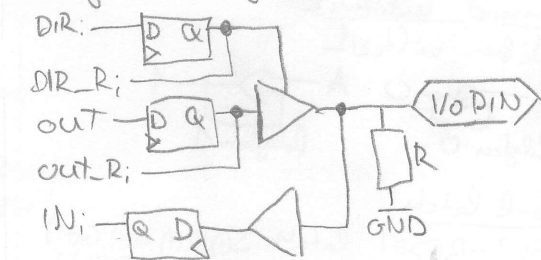
## Busz

- Címbeírás
- Adatbeírás
- Vezérlő busz: Rendszerjel: CLK, RST
- Arbitrációs jel: BUSREQ, BUSACK
- Számvevő jel: READ/WRITE, vagy
- Átírási jel: FRAME TS, TACK, AS, DS
- Megszakítás vezérlő jel: IRQ, SLCK

A BUSZ tulajdonságai: rendszerfüggetlen  
 Mini RISC egyszerű belső busz  
 Kommunikáció: SCIM, & ADAT\_KI, & ADAT\_BE  
 → Vezérlő jel: CLK, RST, WR, RD, IRG  
 A Buszilluszto logika főleg buszról függ

## I/O Periféria

- Adat: cím, regiszter OUT [7:0]
- Adat: cím, regiszter IN [7:0]
- Számvevő regiszter DIR [7:0]

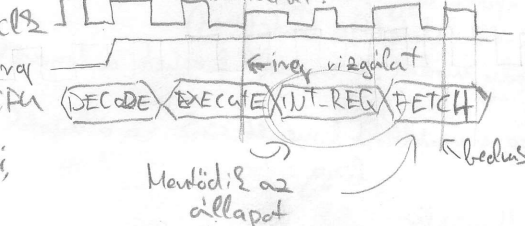


## Megszakítás

- Megszakítás (Interrupts) → Hardware
- Számvevő (Exceptions) → Software
- Interrupts: Normál működés közben  
 Biztonsági leírás  
 Hibátlan utali jelzéssel
- Exceptions: Ritka de bekövetkező események  
 PL: Vannak túlsúlyos, érvénytelen utasítások

IRQ vonal: értesít a perifériáról  
 Megszakítás: letiltatva a megszakítás  
 Vannak NMI & Non-Maskable Interrupts  
 Vezérlő MINDIG érvényes jutnak  
 Megszakításrendszerek változásai

- Egyszintű: Ha már átváltott megszakításra  
 újabbat nem tud fogadni
- Többszintű: Ha már átváltott megszakításra  
 tud újabbat fogadni és  
 folytatni



## Periféria illesztés

- A periféria típusa alapján az igényelt funkciók
- A báziscím kijelölése, a címbeírás megnevezése

## Kommunikáció

UART/USRT: Univerzális aszinkron/ szinkron  
 adatváltó egység  
 Az adat bitet egy start & stop  
 jel veszi kezbe (FRAME)