

# Bevezetés a mérés technikába és jelfeldolgozásba

Tihanyi Attila  
2007. április 24.

# Mikrovezérlők

- 1970 E.M.Hoff javasolja az univerzális vezérlő eszköz kialakítását
- 1971 4004 4040
  - 4 bits 750kHz órajel
  - 8...16 ciklus/utasítás 3 mélységű verem
- 8008 számos utángyártó pl. TEXAS
- Intel 8080 Motorola 6800 Zilog Z80
- Intel 8051 és 8086 irány szétválása

# Információ és feldolgozása

- Információ egysége „bit”
- Jelentése „1”-> Igen „0”-> Nem
- 8 bites sorozata byte tartalma
  - lehet szám -> 0 ... 255
  - lehet karakter -> ‘0’...‘9’ ‘A’...‘Z’ ...
  - lehet analóg érték (ADC DAC)
    - Mintavétel, Kvantálás

# ASCII kódtábla

| Az ASCII<br>kódszavak |      | Felső három bitje |     |     |     |     |     |     |     |
|-----------------------|------|-------------------|-----|-----|-----|-----|-----|-----|-----|
|                       |      | 000               | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Alsó<br>négy<br>bitje | 0000 | NUL               | DLE | SP  | 0   | @   | P   | `   | p   |
|                       | 0001 | SOH               | DC1 | !   | 1   | A   | Q   | a   | q   |
|                       | 0010 | STX               | DC2 | "   | 2   | B   | R   | b   | r   |
|                       | 0011 | ETX               | DC3 | #   | 3   | C   | S   | c   | s   |
|                       | 0100 | EOT               | DC4 | \$  | 4   | D   | T   | d   | t   |
|                       | 0101 | ENQ               | NAK | %   | 5   | E   | U   | e   | u   |
|                       | 0110 | ACK               | SYN | &   | 6   | F   | V   | f   | v   |
|                       | 0111 | BEL               | ETB | '   | 7   | G   | W   | g   | w   |
|                       | 1000 | BS                | CAN | (   | 8   | H   | X   | h   | x   |
|                       | 1001 | HT                | EM  | )   | 9   | I   | Y   | i   | y   |
|                       | 1010 | LF                | SUB | *   | :   | J   | Z   | j   | z   |
|                       | 1011 | VT                | ESC | +   | ;   | K   | [   | k   | {   |
|                       | 1100 | FF                | FS  | ,   | <   | L   | \   | l   | }   |
|                       | 1101 | CR                | GS  | -   | =   | M   | ]   | m   |     |
|                       | 1110 | SO                | RS  | .   | >   | N   | ^   | n   | ~   |
|                       | 1111 | SI                | US  | /   | ?   | O   | _   | o   | DEL |

# Kódok

| Bin    | Gray   | Bin    | Gray   |
|--------|--------|--------|--------|
| • 0000 | — 0000 | • 0101 | — 0111 |
| • 0001 | — 0001 | • 0110 | — 0101 |
| • 0010 | — 0011 | • 0111 | — 0100 |
| • 0011 | — 0010 | • 1000 | — 1100 |
| • 0100 | — 0110 | • 1001 | — 1101 |

# Kódolás tipikus kódok

- Binearis kód

- 1 byte 0 ... 255 1111 1111
- 2 byte 0 ... 65 535 1111 1111 1111 1111
- 3 byte 0 ... 16 777 215
- 4 byte 0 ... 4 294 967 295

$$v = \sum_i c_i 2^i$$

# Hexadecimális ábrázolás

- 1 byte 0 ... 0FFH                      0 ... 0xff
- 2 byte 0 ... 0FFFFH                    0 ... 0xffff
- 3 byte 0 ... 0FFFFFFFH                0 ... 0xffffffff
- 4 byte 0 ... 0FFFFFFFFFH            0 ... 0xffffffff

$$v = \sum_i k_i 16^i$$

# BCD ábrázolás

- 1 byte 0 ... 99H                      0 ... 0x99
- 2 byte 0 ... 9999H                  0 ... 0x9999
- 3 byte 0 ... 999999H              0 ... 0x999999
- 4 byte 0 ... 99999999H          0 ... 0x99999999

$$v = \sum_i k_i 10^i$$



# Negatív számok

- Komplement kódú ábrázolás
  - Írjuk le külön 1 biten a szám előjelét
    - ➔ létezik +0 és -0 ez nem biztos, hogy jó
- 2-es komplementes kódú ábrázolás
  - Legmagasabb helyiértékű bit az előjel
    - $H_a == 1$  ➔ negatív a szám
    - $H_a == 0$  ➔ pozitív a szám

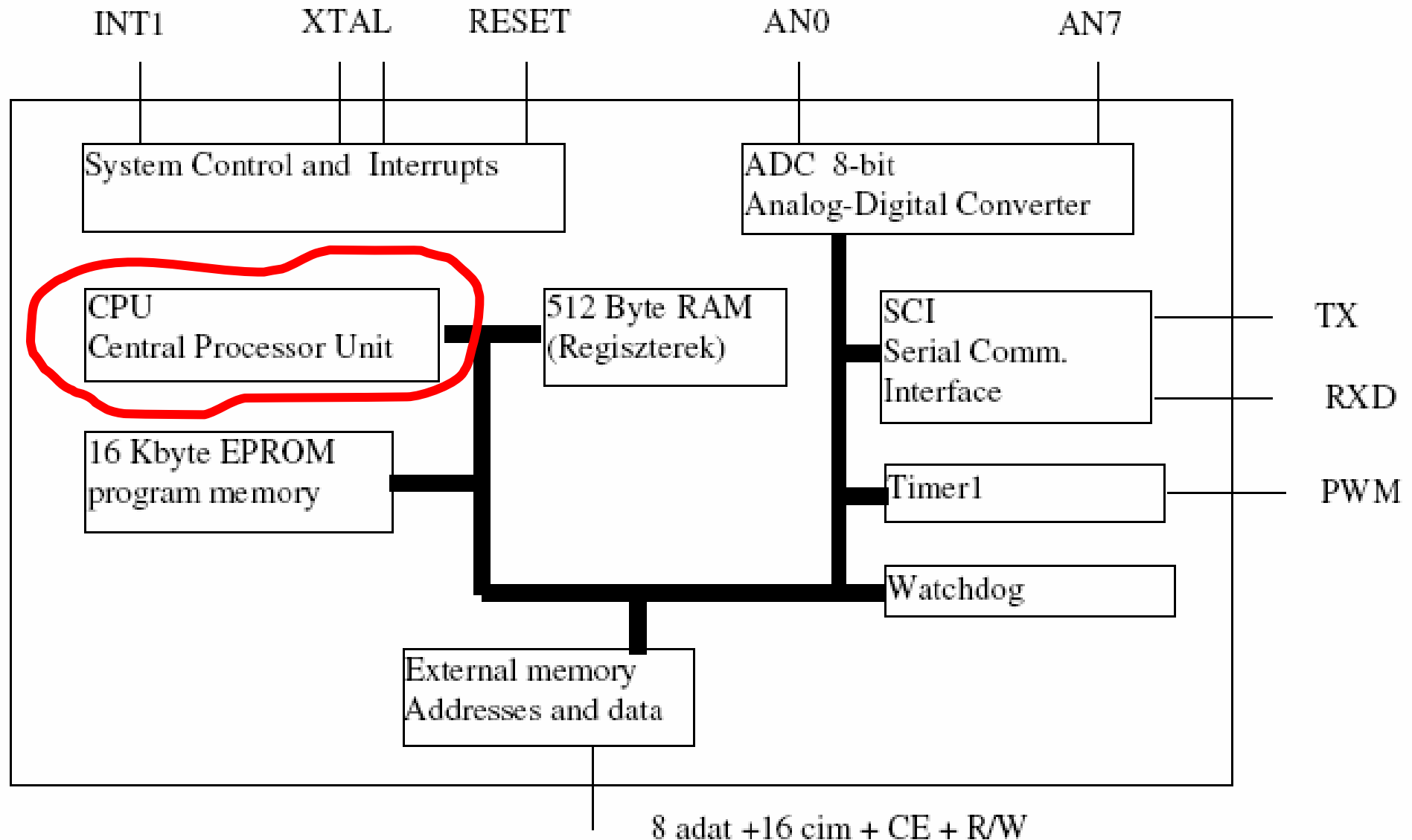
# Komplementens kódok

- Binearis kód
  - 1 byte -128 ... 127
  - 2 byte -32 768 ... 32 767
  - 3 byte -8 388 608 ... 8 388 608
  - 4 byte -2 147 483 648 ... 2 147 483 647

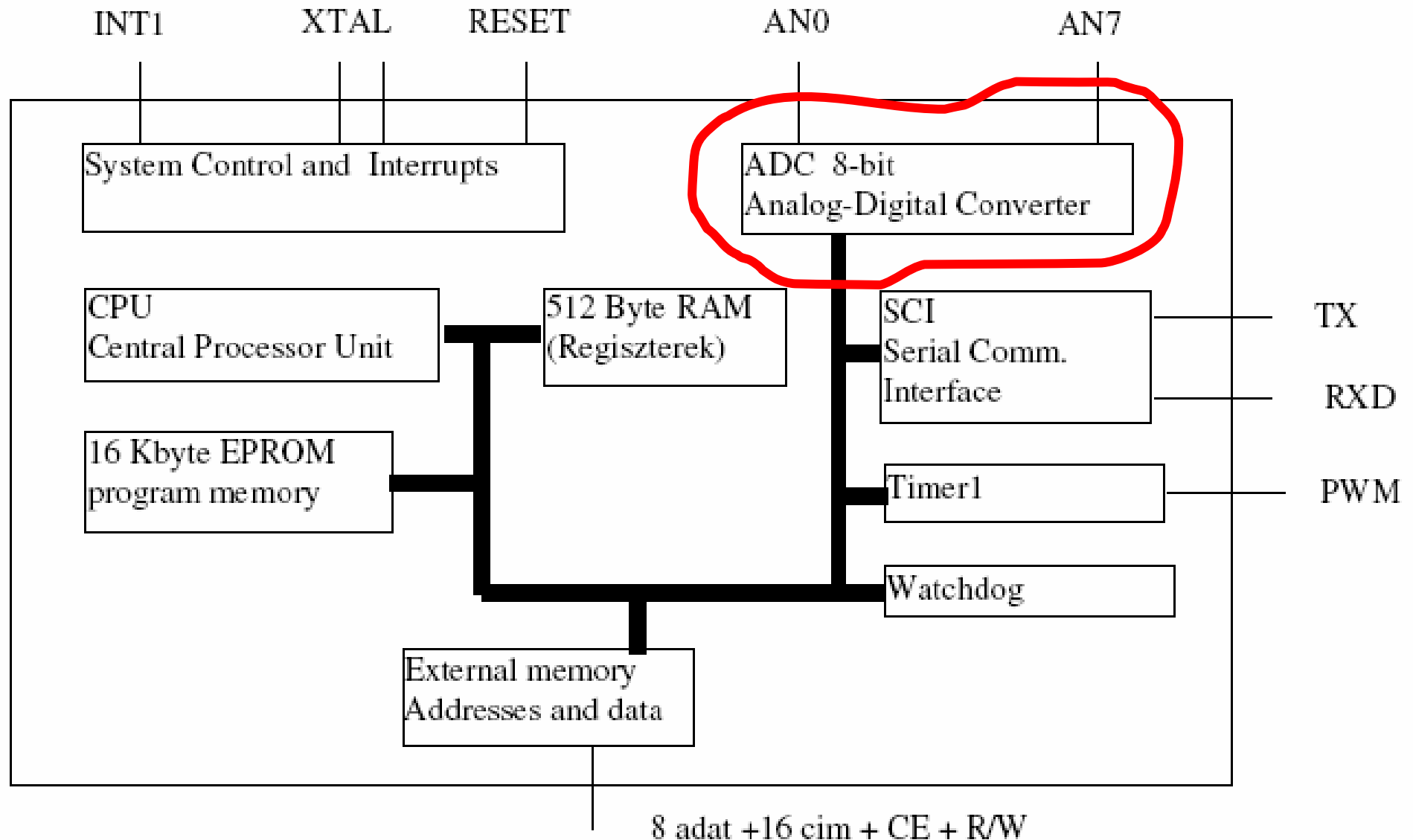
# Számok

# Gyakorlás

# TEXAS TMS370

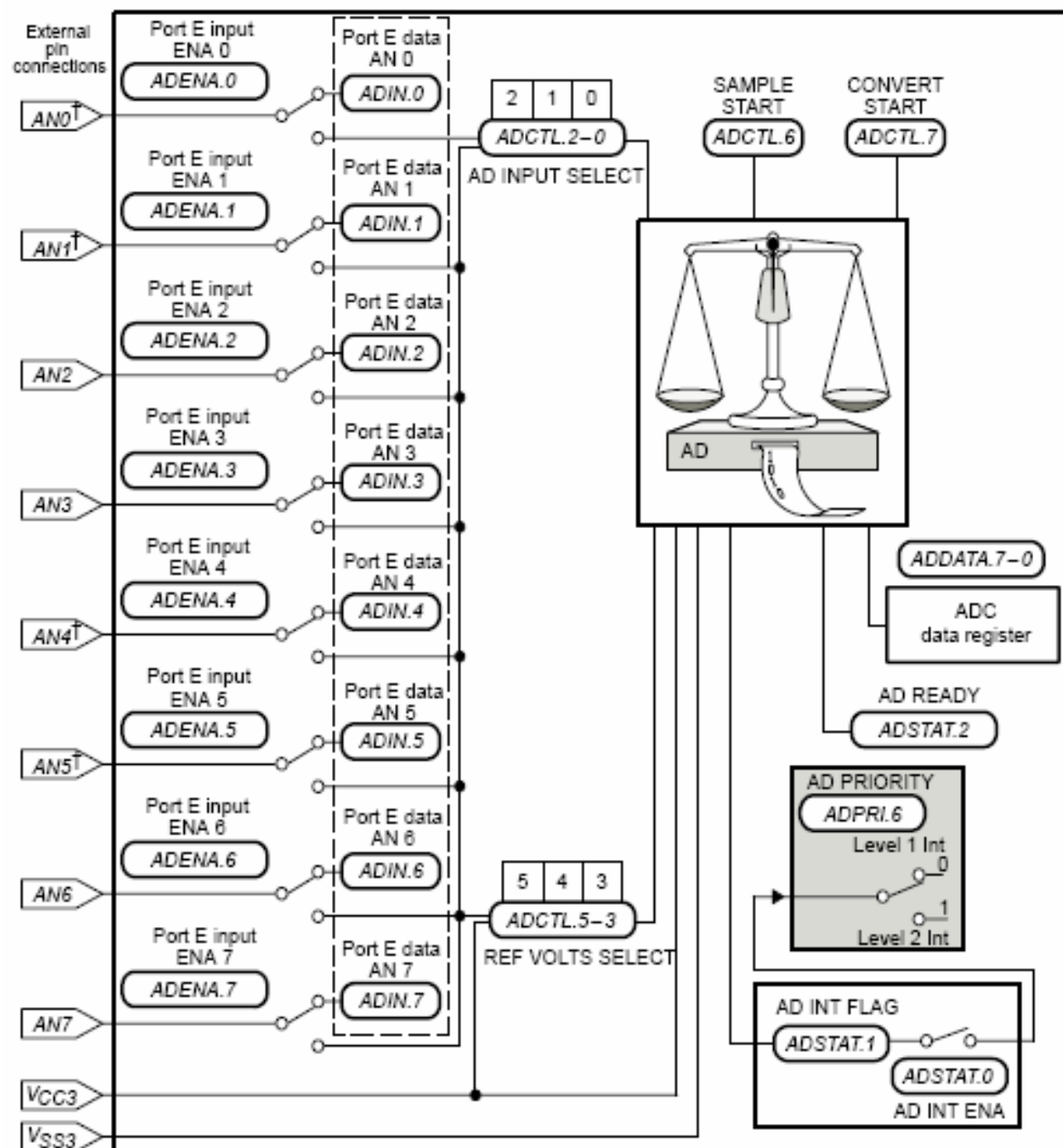


# TEXAS TMS370



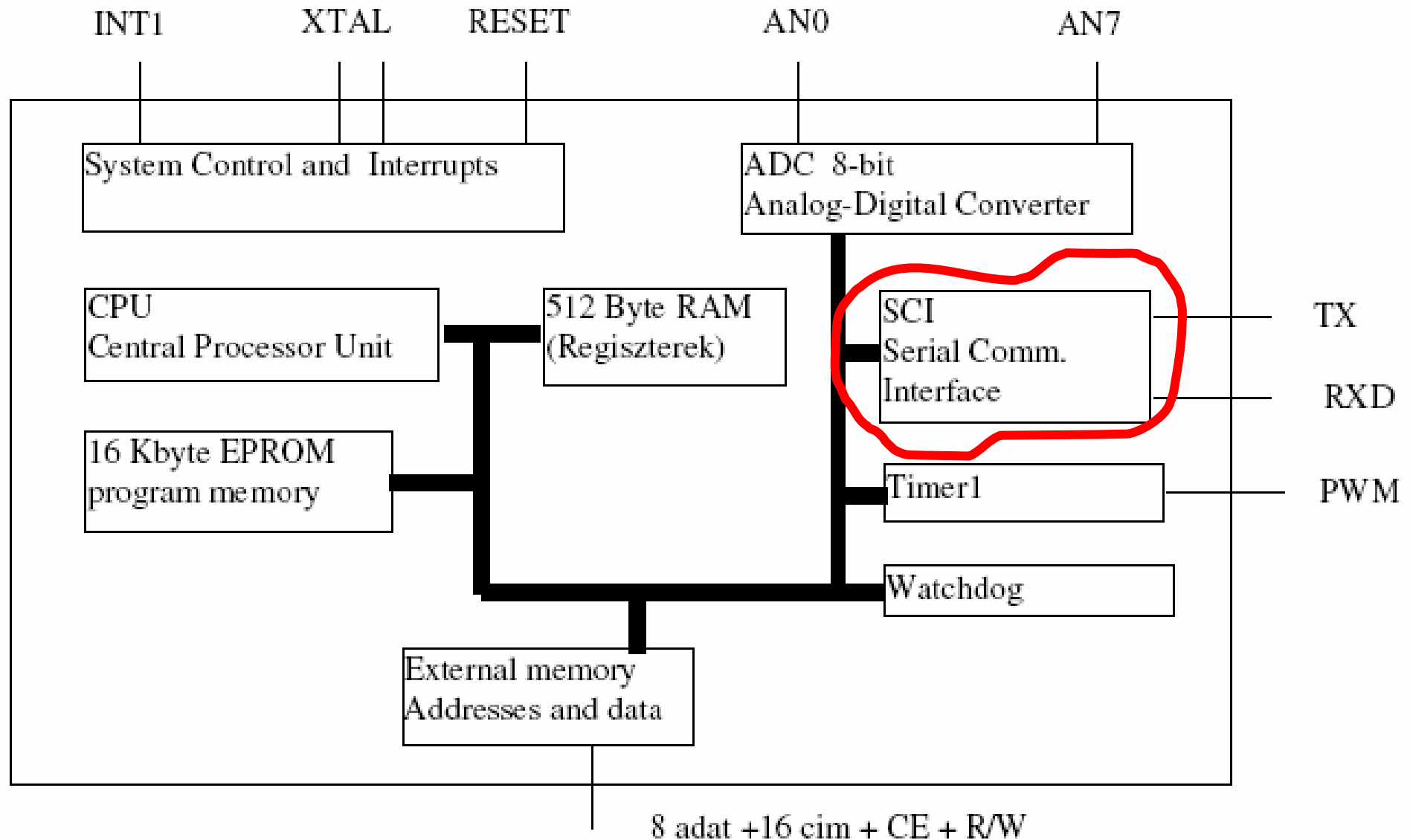
# Analóg bemenet(ek)

- 8 bites konverter
- Max 5V bemeneti feszültség
  - $5V/255 = 19,6mV$
- Intervallum felezéses üzemmód
- Kb 100 ks/s sebesség

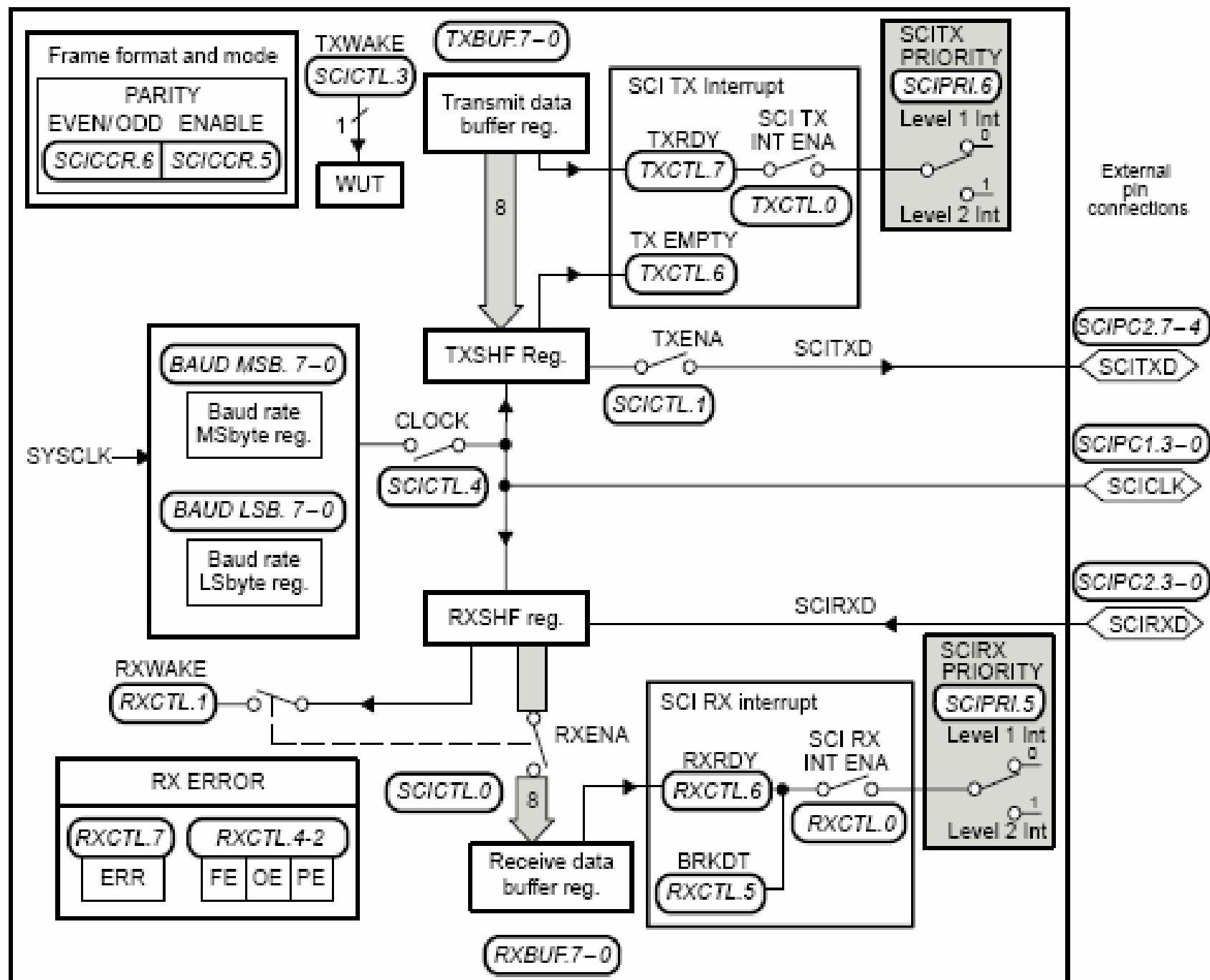


† These pins are not implemented in the TMS370C4x 40-pin device.

# TEXAS TMS370

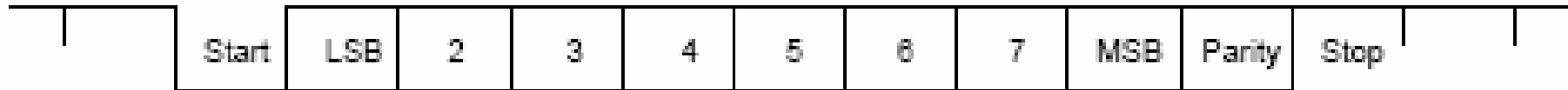






Note: SCI1 registers are described in detail in Section 9.8.1 beginning on page 9-22.

# Aszinkron soros kommunikáció



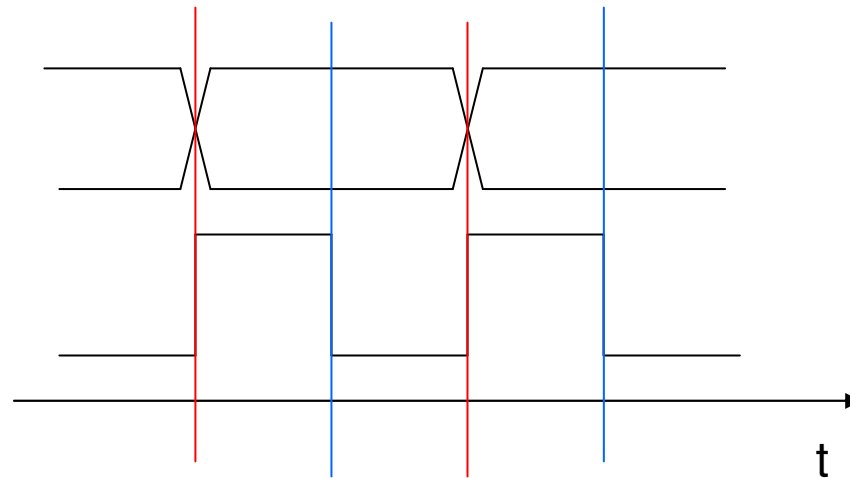
Idle Line Mode  
(Normal Nonmultiprocessor Communications)

- Start bit **mindig 1**
- Adatok alacsony magas sorrendben **5;6;7;8;**
- Paritás **N;M;S;O;E;**
- Stop bit **1;1,5;2;**
- Névleges sebesség
  - **50, 100, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, ...**

# Synchron serial communication

- SPI

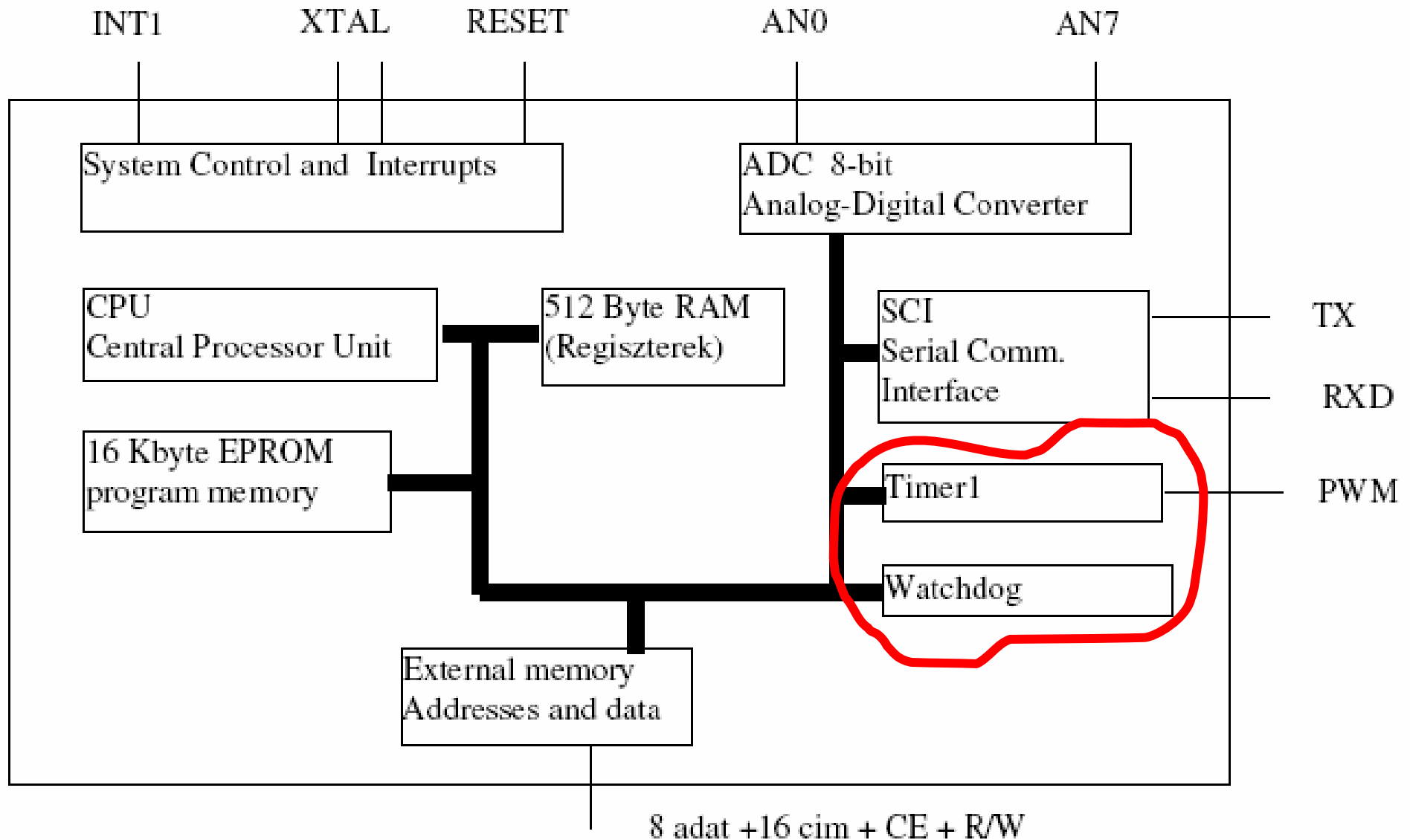
- 1 adat bemenet
- 1 adat kimenet
- 1 órajel
- 1 szelekt jel



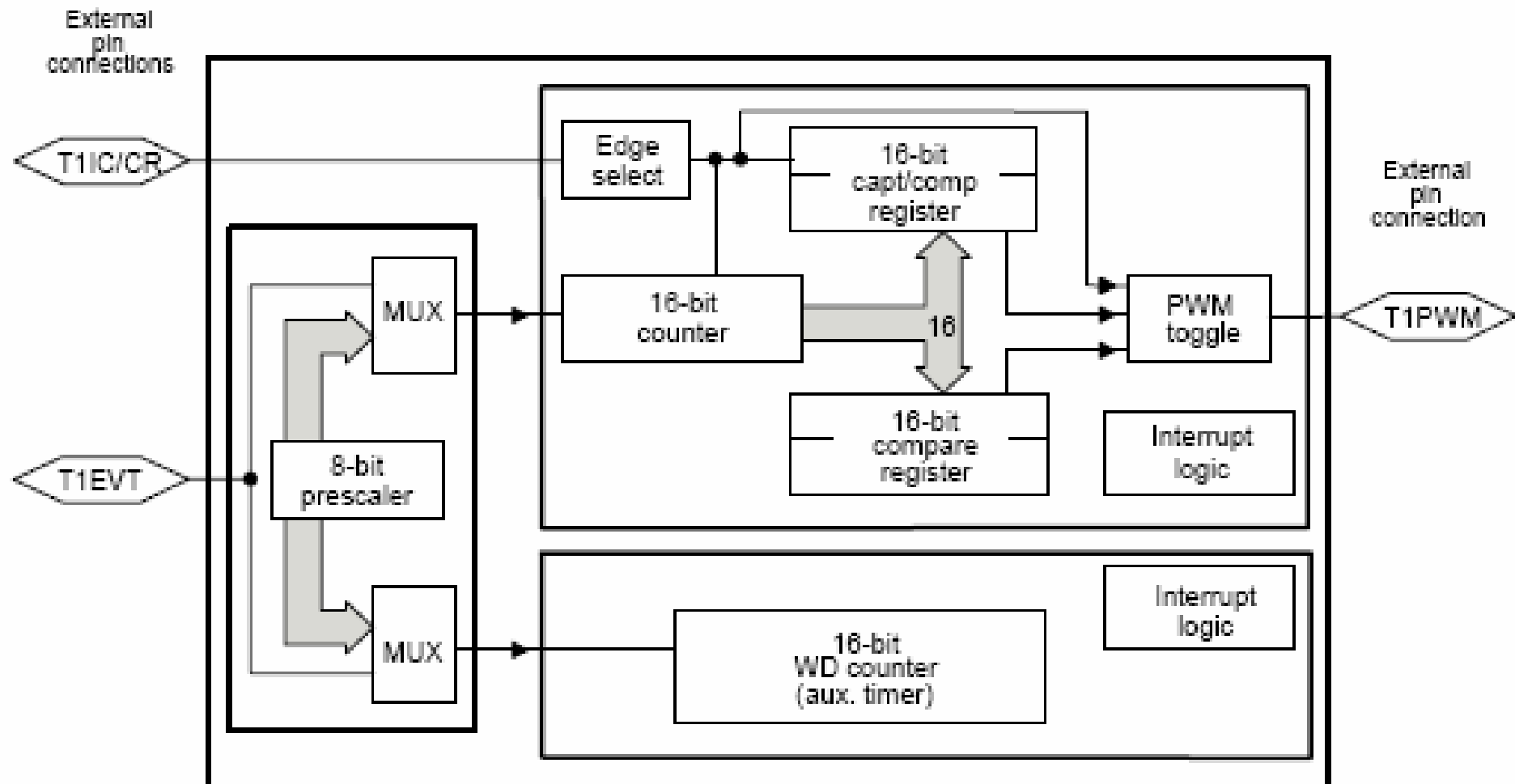
- I2C

- 1 adatjel kétirányú
- 1 órajel

# TEXAS TMS370

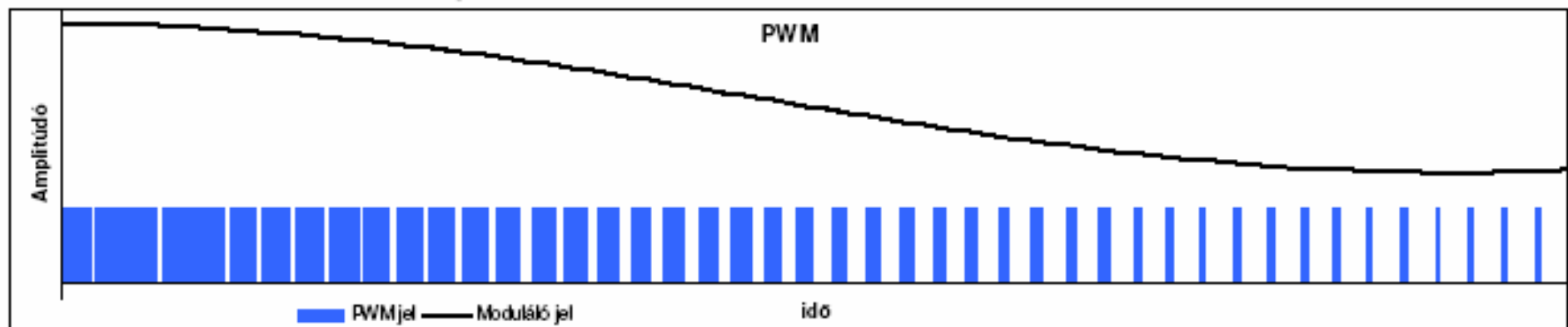


# Timer



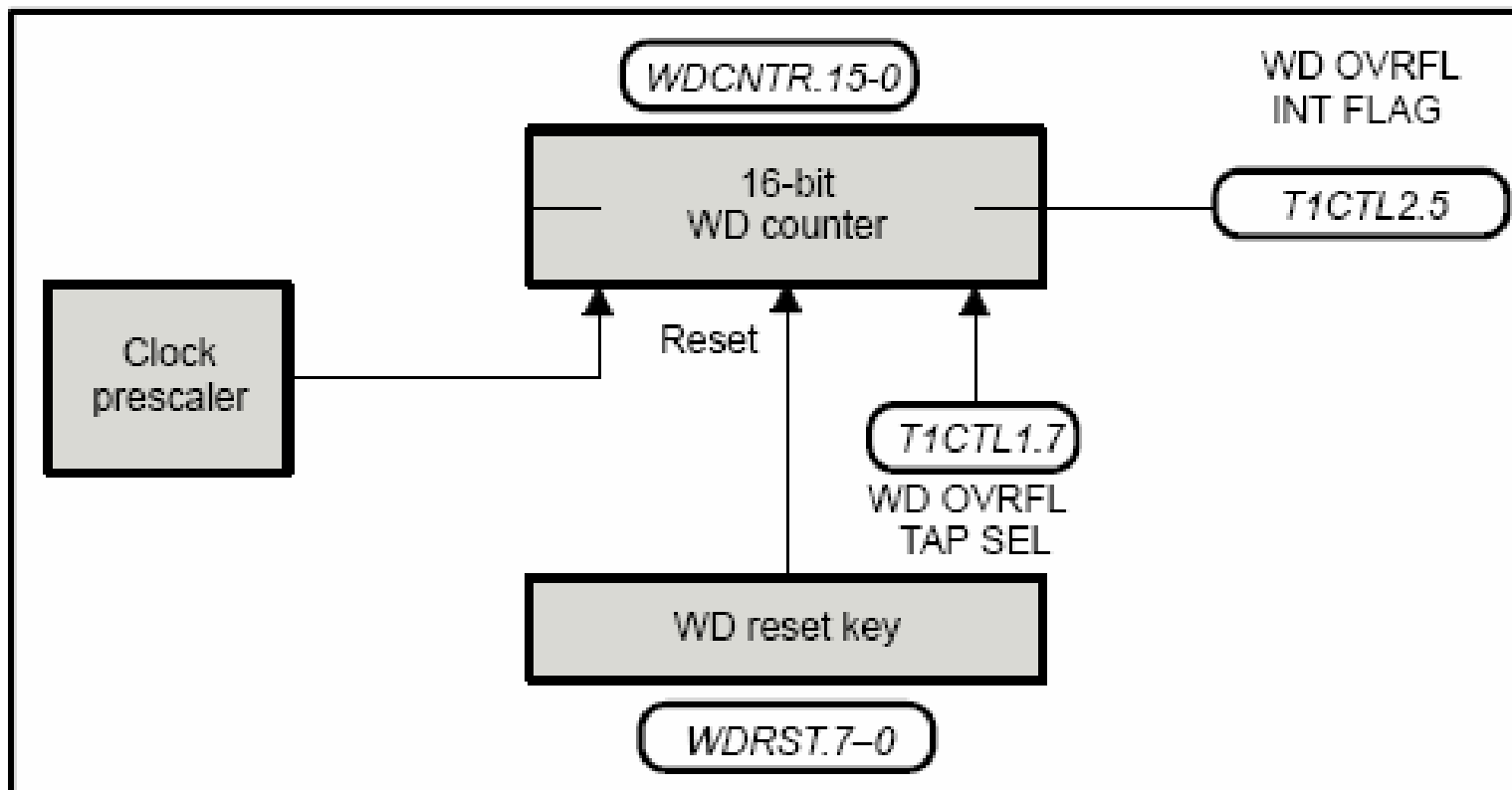
# Pulse Width Modulation

- Egyszerű DAC megoldás

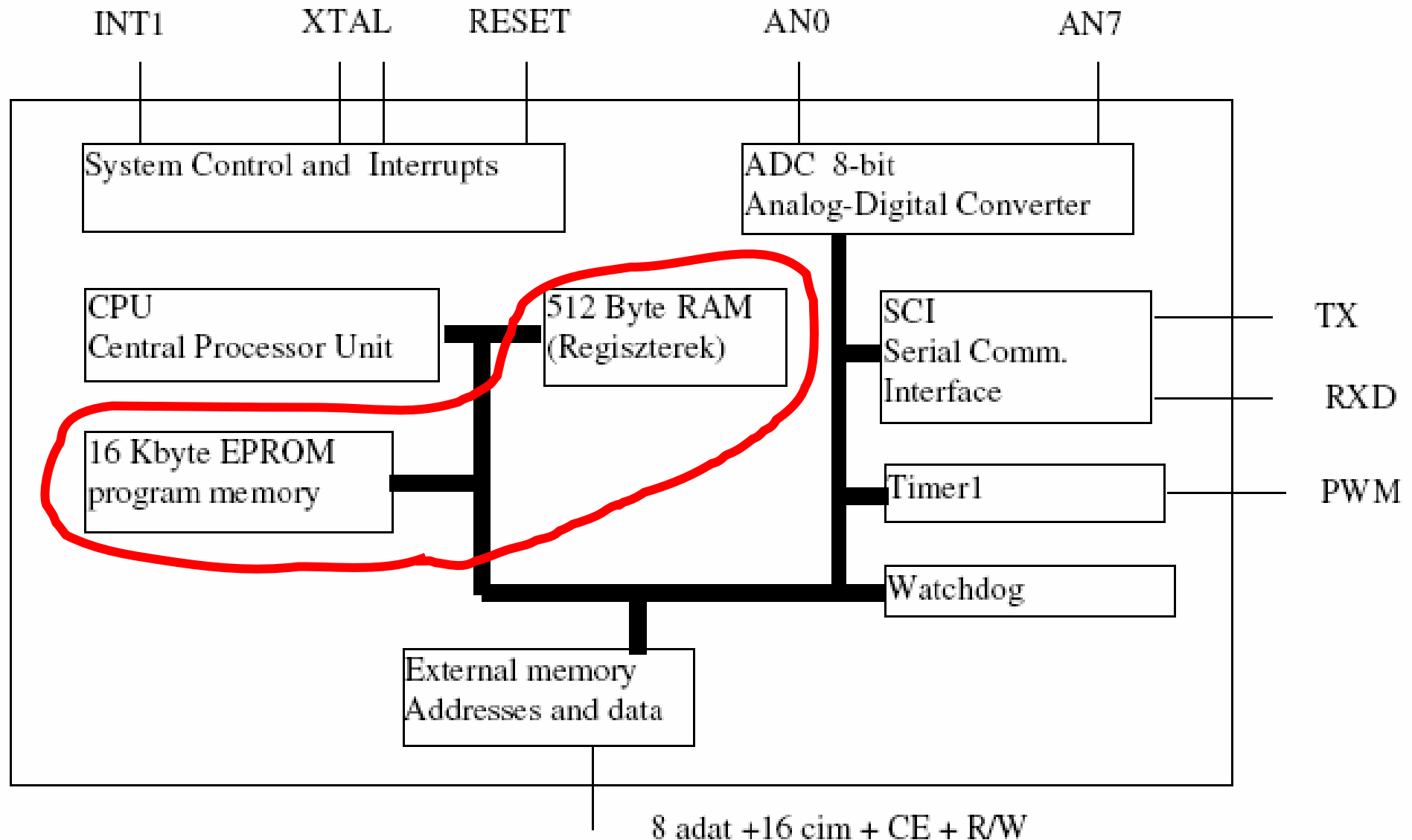


# WATCHDOG

- Újraindítható monostabil multivibrátor
- Program futásának ellenőrzésére alkalmas
- Biztonsági megoldások

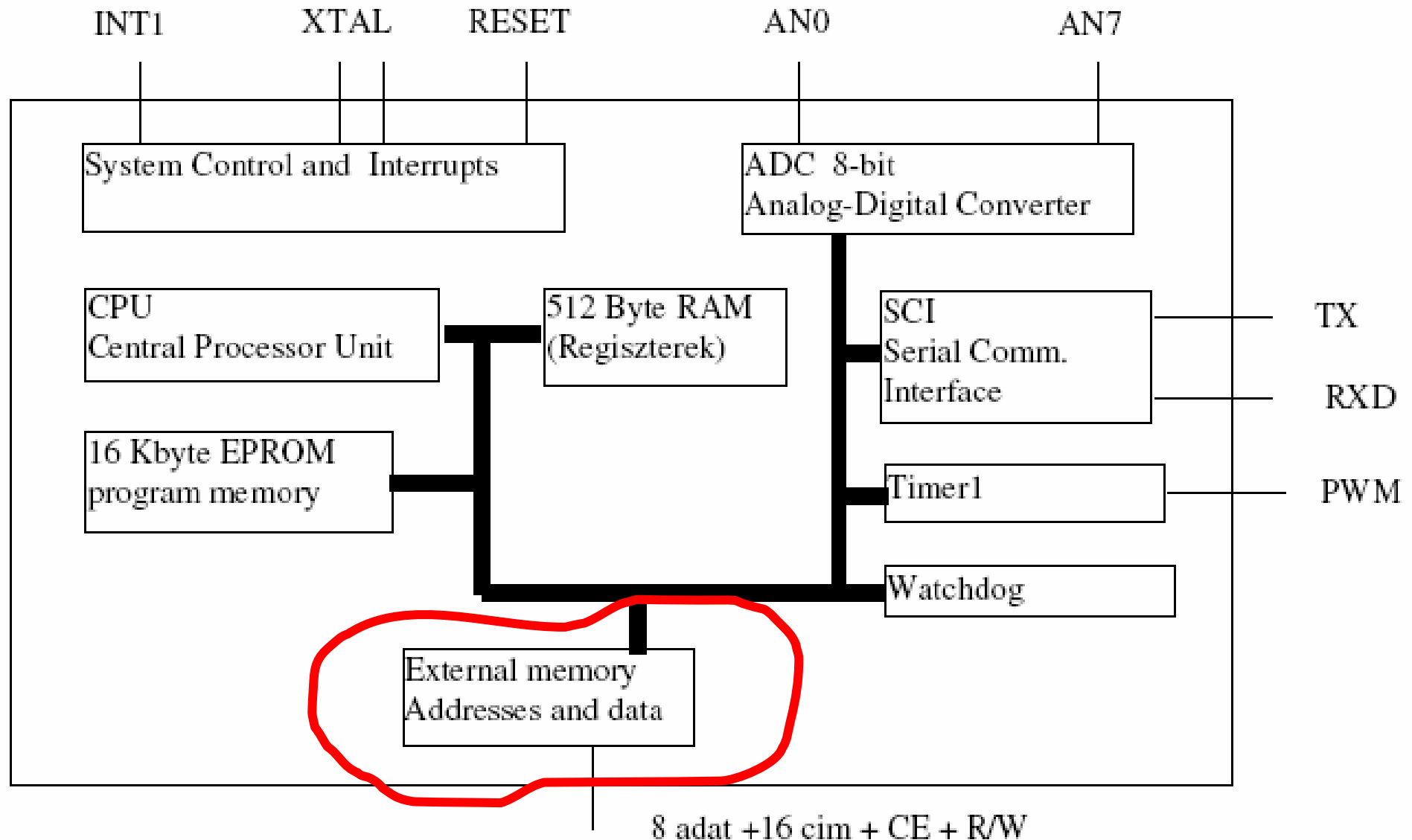


# TEXAS TMS370

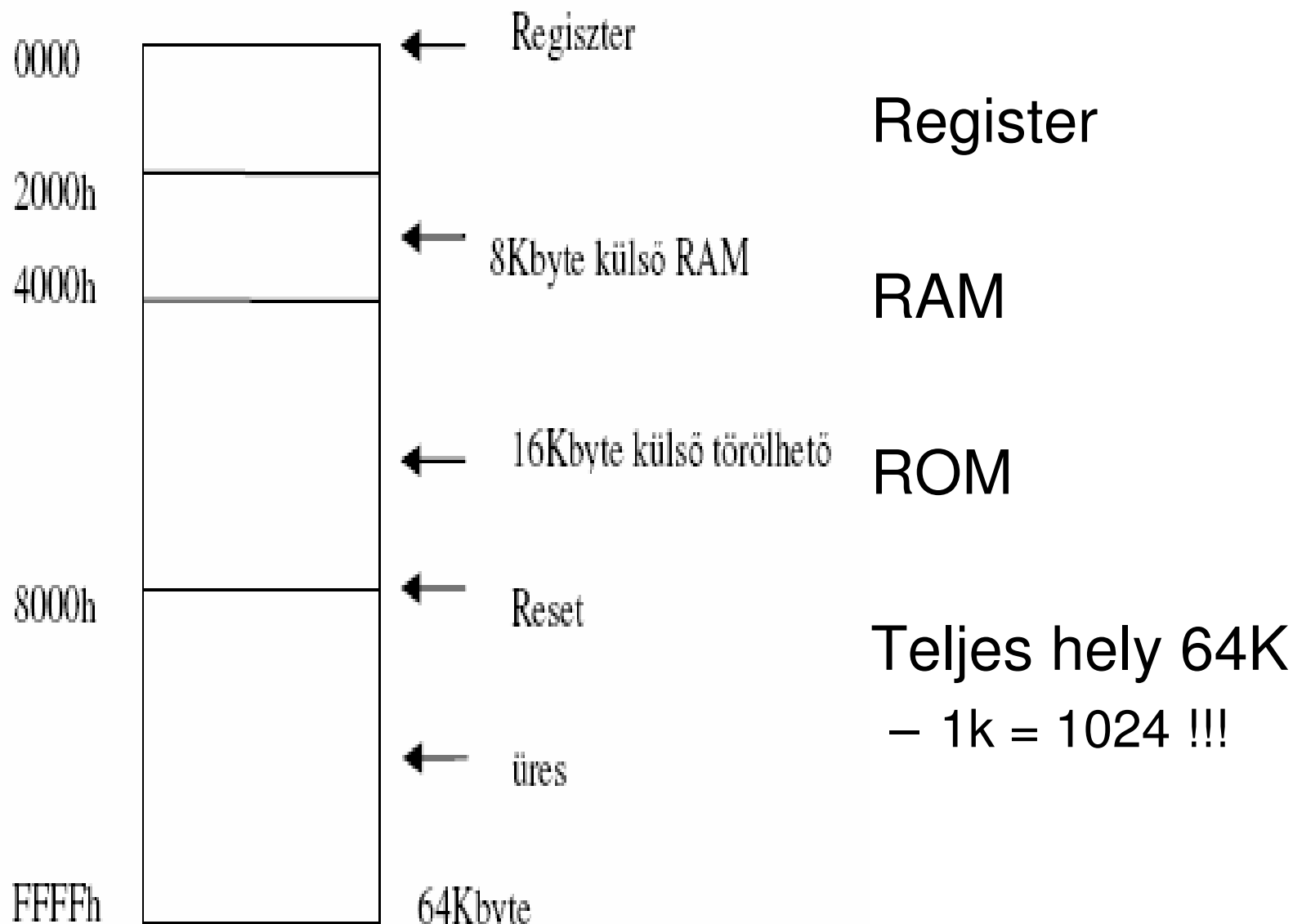




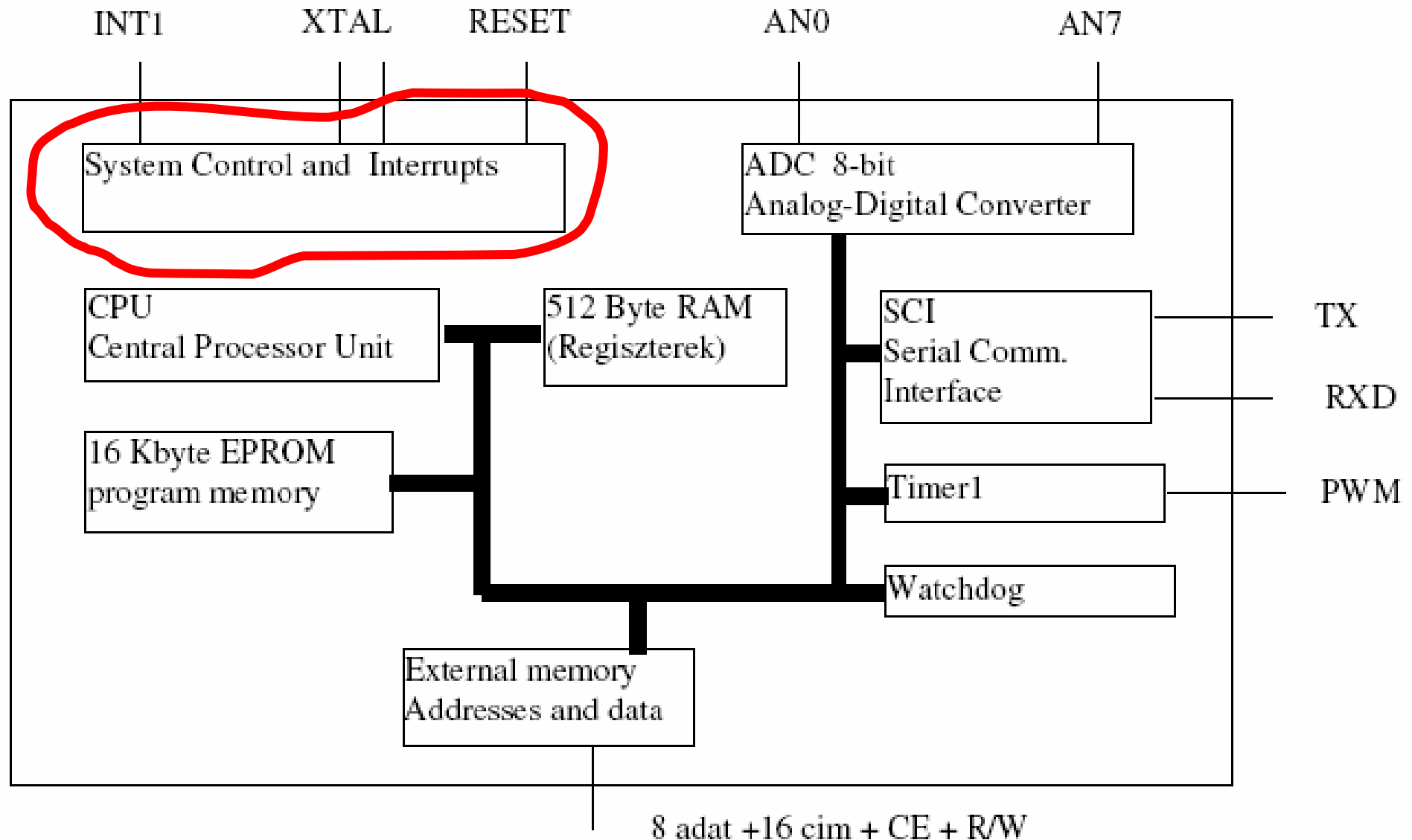
# TEXAS TMS370



# Külső memóriák



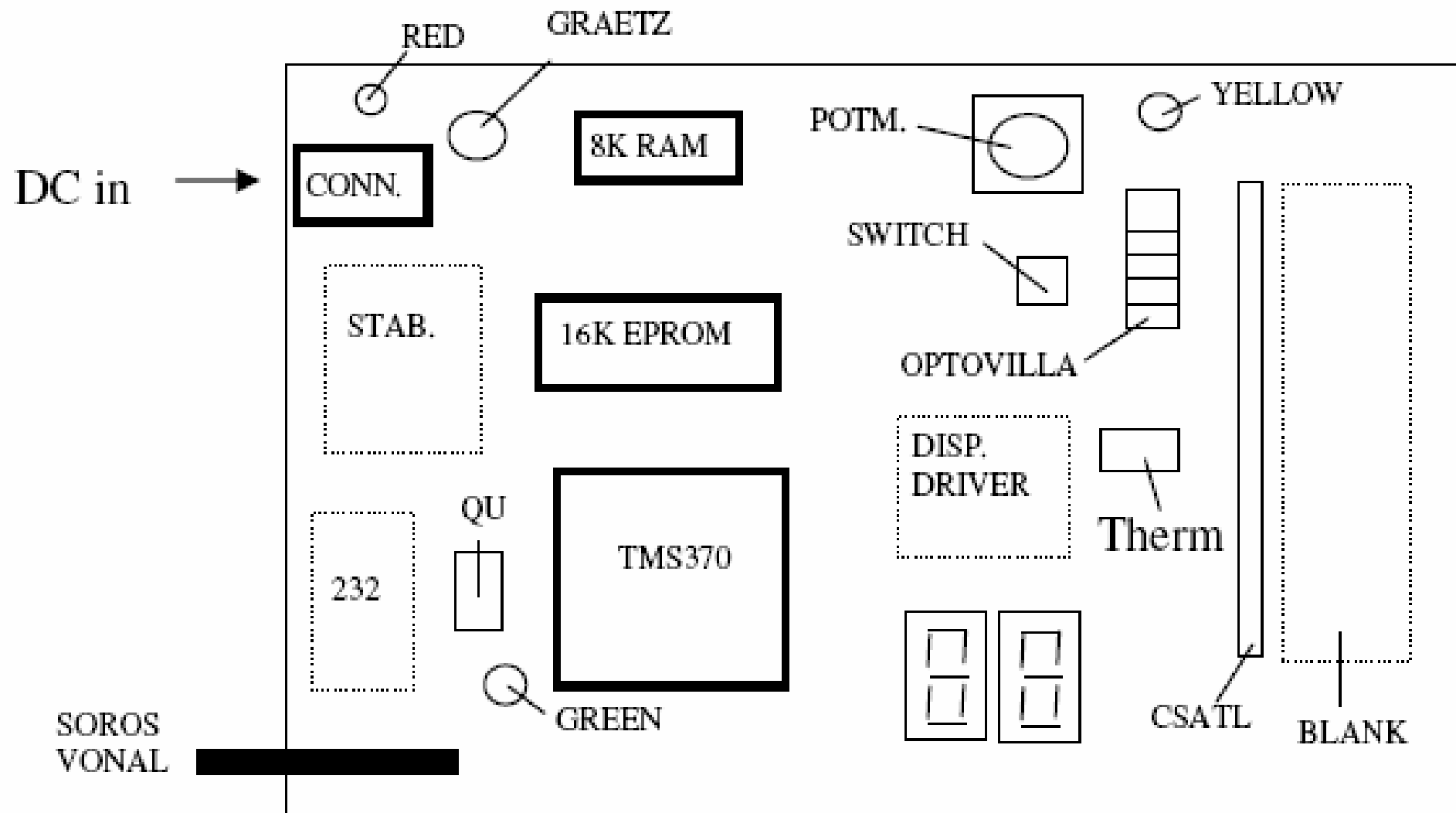
# TEXAS TMS370



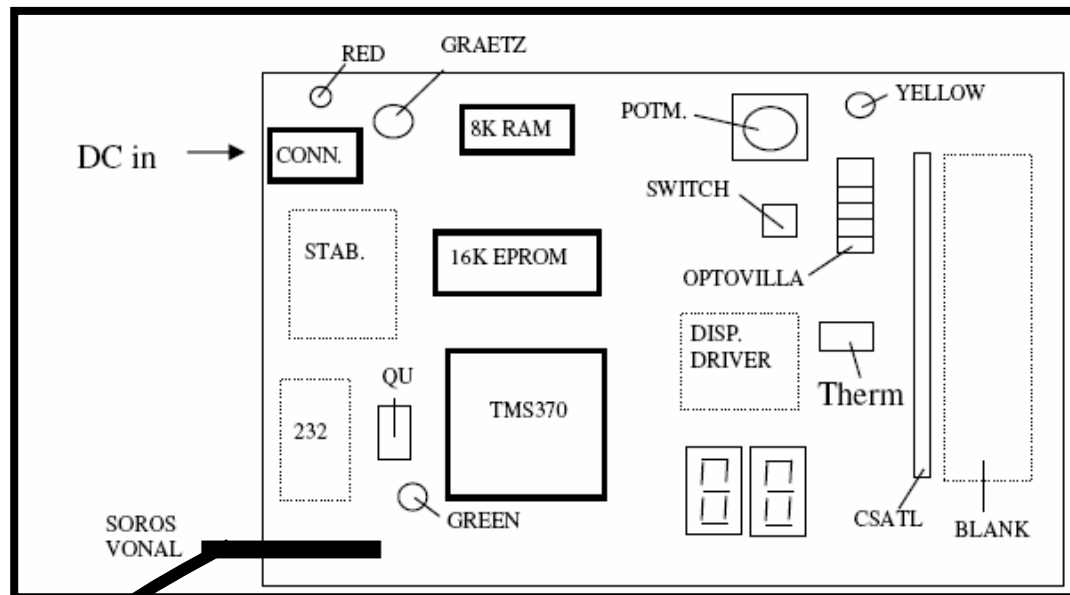
# Interrupt

- Program megszakítás
  - Külső forrásból
  - Belső forrásból
  - Szoftver
- Szükséges műveletek
  - Mentés, IT tiltás, alprogram hívás (processor)
  - Mentés (felhasználó)
  - IT kód
  - IT forrás megszüntetése
  - Visszaállítás (felhasználó)
  - Visszatérés, IT engedélyezés

# Fejlesztői környezet

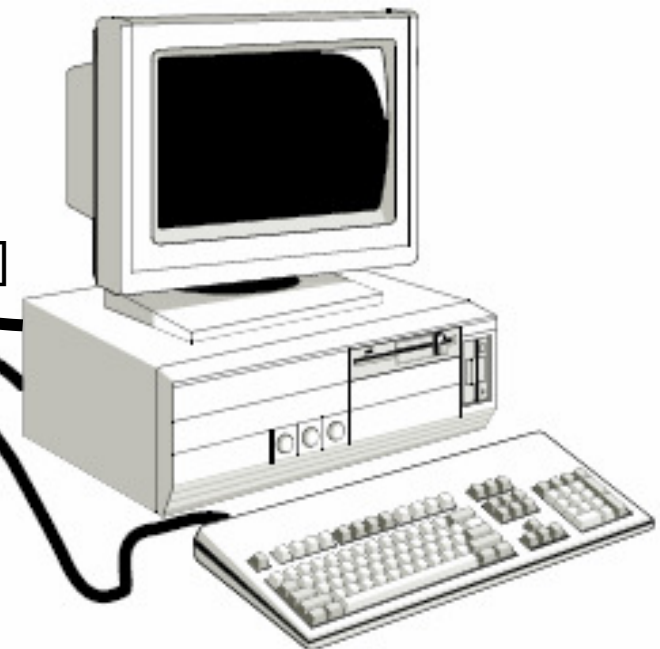


# Összeállítás



C:\TMS370\TMS94.exe

Com[1;2]

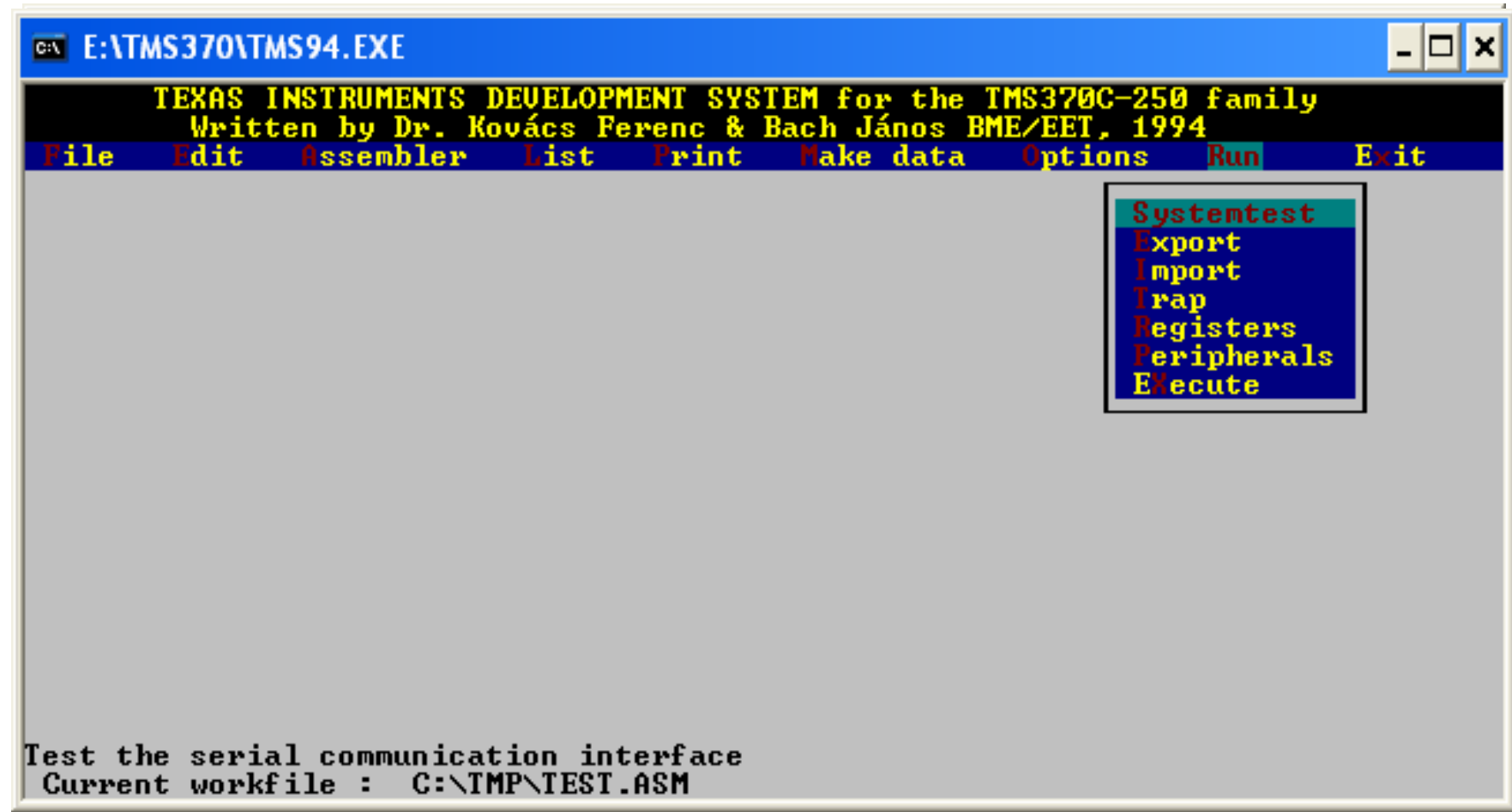


# Működik?

- Indítsuk el TMS94.exe programot. Ezt követően:
  - először definiálni kell azt file-t, amin dolgozni akarunk,
  - ezután OPTION menüpontban nyomjunk egy „U”-t,
  - adjuk meg a COM[1;2]-öt,
  - ellenőrizzük, hogy kap-e tápfeszültséget a mikrogép (vagyis világít-e a piros LED),
  - ellenőrizzük, hogy fut-e a monitor a mikrogépen, vagyis villog-e a zöld LED (ha nem, nyomjuk meg a RESET gombot)
  - ezután ellenőrizzük a mikrogép és a PC kapcsolatát (RUN, Communication), aminek hatására egy villanás után semmi új nem íródik ki; hiba esetén „no communication” jelenik meg.

**IGEN!**

# TMS94.exe





# Program készítés

MEMORY

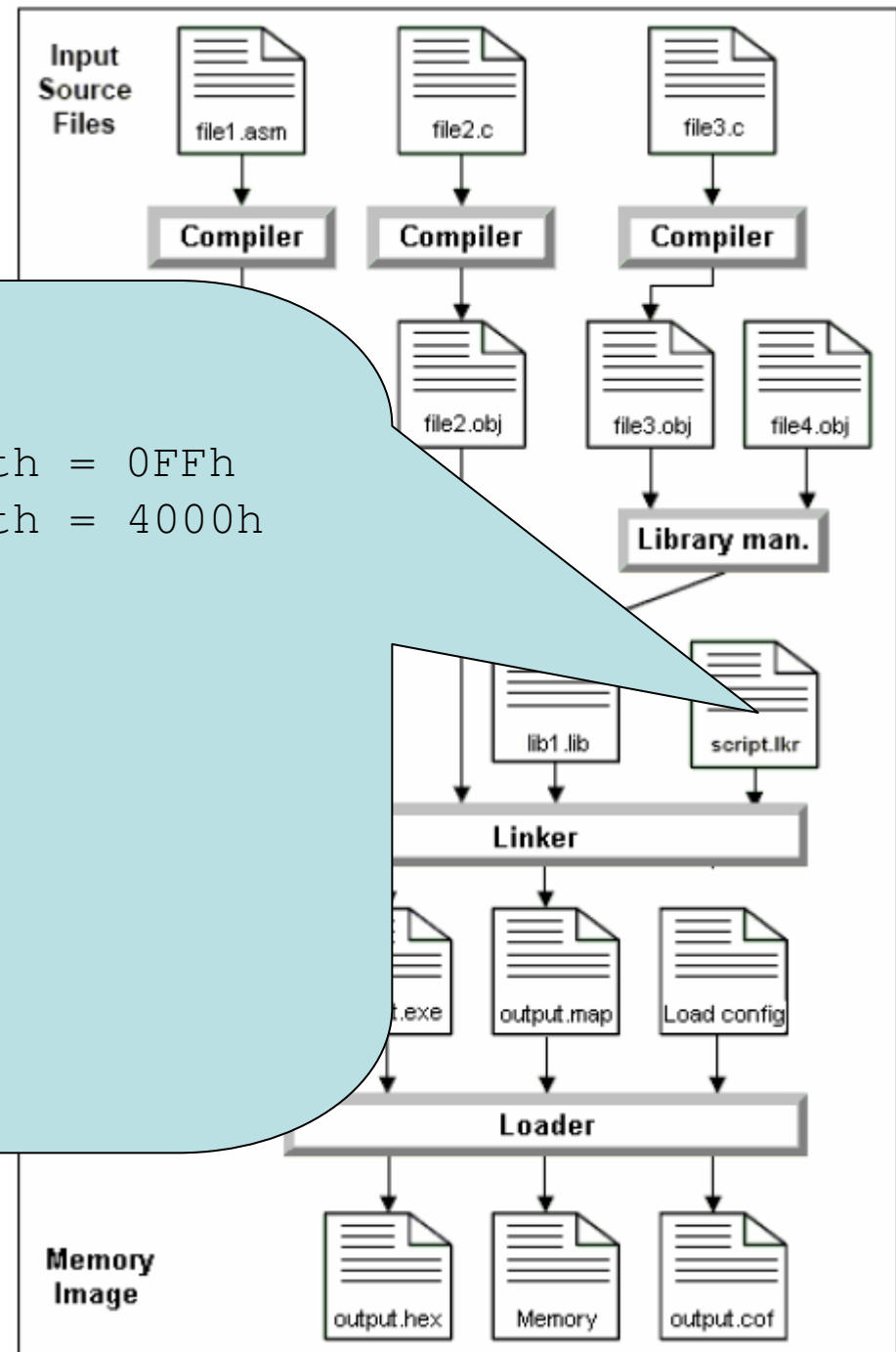
```
{  
  RAM: origin = 000h      length = 0FFh  
  ROM: origin = 02000h    length = 4000h  
}
```

SECTIONS

```
{  
  .text: > ROM  
  .data: > ROM  
  .bss:  > RAM  
}
```

terkep me

Ez a végrehajtható kód



# Lépések

- EDIT módban az assembler-nyelv program beírása,
- ASSEMBLER módban fordítás; hiba esetén vissza EDIT-be és javítás,
- „no error” esetén MADE módban az object kód elállítása,
- RUN módban „Export” utasítással a program letöltése a RAM 2000h címére,
- „Trap” módban a kívánt megállási cm beírása (ha ráfut a program, visszaugrik a monitorba),
- „Execution” módban a program elindítása, célszeren 2000h címről,
- Előírásszerű trap-re futás után „Registers” vagy „Import” utasítással a memóriák visszaolvasása táblázatosan, hexadecimális formában.

# ASM formátum

Megjegyzés

Címke

Közvetlen operandus  
Numerikus konstans

- ; ----- KERET-PROGRAM -----
- START:      mov #60h,B
- ldsp                      ; stack kezdete REGfile-ban, R60
- DINT                      ; Utasítás mnemonik
- call INIT                      ; inicializálás
- ; ---- hallgatói program kezdete:
- TRAP 0                      ; Közvetlen operandus
- .end

Címke hivatkozás

Közvetlen operandus

Forrás vége !!!

# LST formátum

Sorszám

Elhelyezés számláló

```
75 2000                                .text  2000h    ; a program 2000h-nal indul
77                                     ;-----
78 2000  '8c201f                        br  START      ; should be branch, 3 byte!
79 2003    00                          .byte  0,0,0,0,0,0,0,0,0,0,0,0,0,0
    2004    00
    2005    00 same ...
81                                     ; itt olvashatok le az interrupt vektorok
82                                     ; a program es innen tovabb az illetekes cimkere
84 2010  '8c20f9                        br  TIM2INT     ; timer2 interrupt jump
85 2013  '8c20f7                        br  TIM1INT     ; timer1, address=2013h
86 2016  '8c20fb                        br  2016h      ; address should be=2016h
87 2019  '8c20fa
88 201c  '8c20f8
91                                     ;----- PROGRAM -----
93 201f    5260                        START:  mov  #60h,B
94 2021    fd                          ldsp          ; stack kezdete REGfile-ban
95 2022    f000                        DINT
96 2024  '8e2069                        call  INIT      ; inicializalas

143 2069    f7113e                     INIT:  mov  #11h,P03E ; SPISOMI/SIMO=outputs
144 206c    f7013d                     mov  #01,P03D  ; SPICLK=gen. purp. output
145 206f    f7015d                     mov  #01,P05D  ; SCICLK=gen. purp. output
```

Programkód relatív címmel

Programkód közvetlen operandussal

# LST formátum

- Elhelyezés számláló
  - Abszolút című forrás
  - Relatív című fordítás
- Utasításkód
- Közvetlen operandus
- Vagy cím
  - Abszolút
  - Relatív

# Utasítások

- NOP ;nincs művelet
- MOV Rs,Rd ;mozgatás Rs-ből ➔ Rd-be
  - MOV A,R06
  - MOV #2,R07
  - MOV #44H,R08
- .equ ;szöveg helyettesítő (#define)
  - C .equ R02

# Műveletvégzés

- Összeadás
- ADD       $R_s, R_d$       ;  $R_s + R_d \rightarrow R_d$ 
  - Összeadás, átvitel (túlcsordulás) kezelése
- ADC       $R_s, R_d$       ;  $R_s + R_d + C_y \rightarrow R_d$
- Kivonás
- SUB       $R_s, R_d$       ;  $R_d - R_s \rightarrow R_d$
- CMP       $R_s, R_d$       ; nincs eredmény
  - Átvitel (alulcsordulás) kezelése
- SBB       $R_s, R_d$       ;  $R_d - R_s - 1 + C_y \rightarrow R_d$

# Műveletvégzés II

- Összeadás
- INC     Rd     ;  $Rd + 1 \rightarrow Rd$
- INCW   #iop8,Rp ;  $Rp + \#iop8 \rightarrow Rp$ 
  - Összeadás, átvitel (túlcsordulás) kezelése
- Kivonás
- DEC     Rd     ;  $Rd - 1 \rightarrow Rd$ 
  - Átvitel (alulcsordulás) kezelése
- Kinulláz
- CLR     Rd     ;  $0 \rightarrow Rd$



# Műveletvégzés III

- Szorzás
- MPY       $R_s, A$       ;  $R_s * A \rightarrow (A:B)$
- Osztás (egészosztás)
- DIV       $R_s, A$       ;  $(A:B) / R_s \rightarrow A; B$ 
  - A eredmény
  - B maradék

# Logikai műveletek

- AND     Rs,Rd     ;  $R_s \& R_d \rightarrow R_d$
- OR      Rs,Rd     ;  $R_s | R_d \rightarrow R_d$
- XOR     Rs,Rd     ;  $R_s \wedge R_d \rightarrow R_d$
- INV      Rd        ;  $\sim R_d \rightarrow R_d$
- CMPBIT name     ;  $\sim \text{name} \rightarrow \text{name}$

– Nem állít Cy flag-et

# Vezérlés átadás

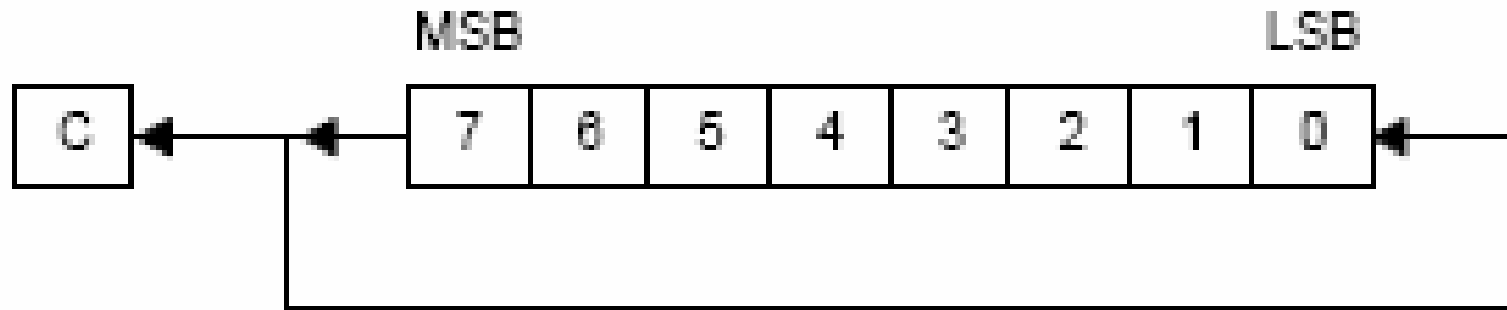
- JMPL címke16 ; feltétel nélküli ugrás
- JMP címke8 ; feltétel nélküli ugrás
- Feltételes ugrások
- JNZ címke8 ; jump if no zero
- JZ címke8 ; jump if zero
- JNC címke8 ; jump if no carry
- JC címke8 ; jump if carry
- DJNZ Rn,címke8 ; decrement and jump no zero
- ...

# Vezérlés átadás II

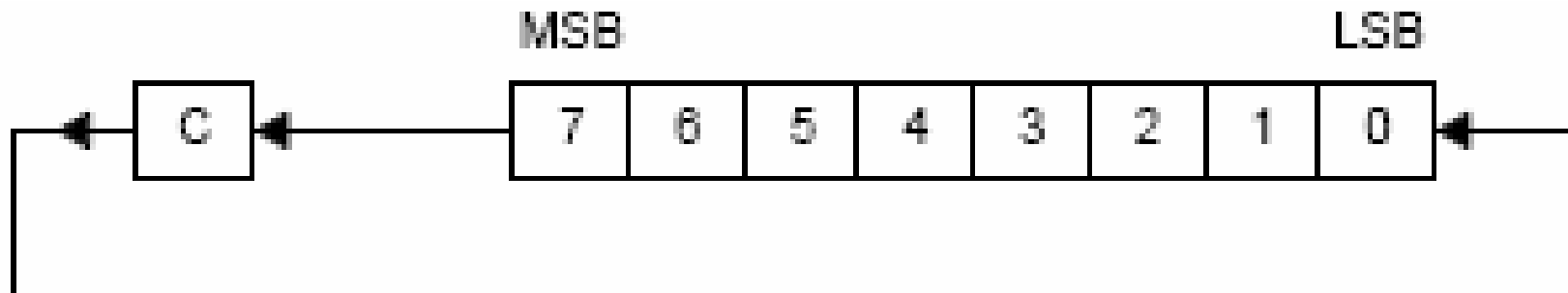
- Subroutine
  - CALL címke ; feltétel nélküli hívás
  - CALLR címke ; feltétel nélküli hívás
    - Visszatérési cím a stack-be
- Visszatérés
  - RTS ;return form subroutine
  - RTI ;return from interrupt
- Stack kezelés
  - PUSH d ;d → (SP)
  - POP d ;(SP) → d

# Rotate

- RL Rd ;rotate left

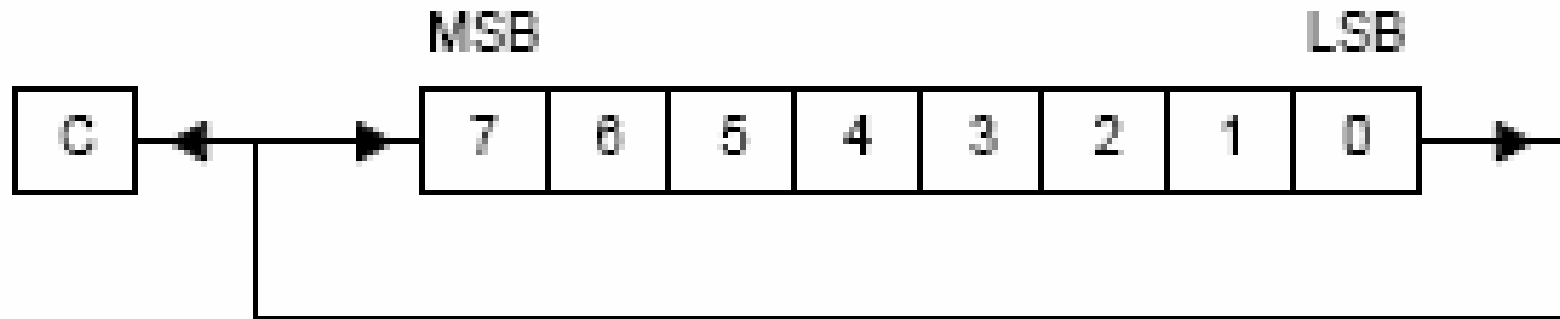


- RLC Rd ; Rotate Left Through Carry

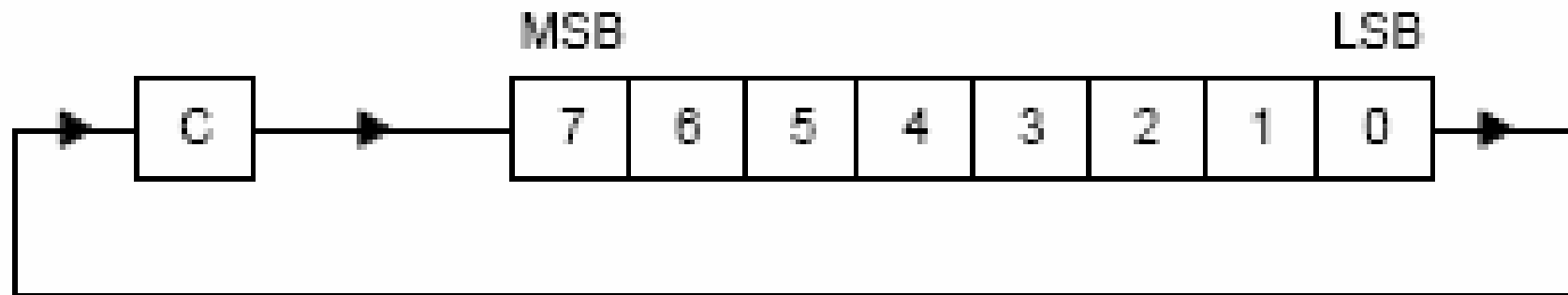


# Rotate

- RR Rd ;rotate left



- RRC Rd ; Rotate Left Through Carry



# Bitműveletek

- SBIT0      name            ; 0 → name
- SBIT1      name            ; 1 → name
  
- JBIT0      name,címke8
- JBIT1      name,címke8
- BTJO      s1,s2,címke8        ;s1 & s2
- BTJZ      s1,s2,címke8        ;s1 & s2
  
- SETC                            ; 1 → Cy
- CLRC                            ; 0 → Cy

# Speciális műveletek

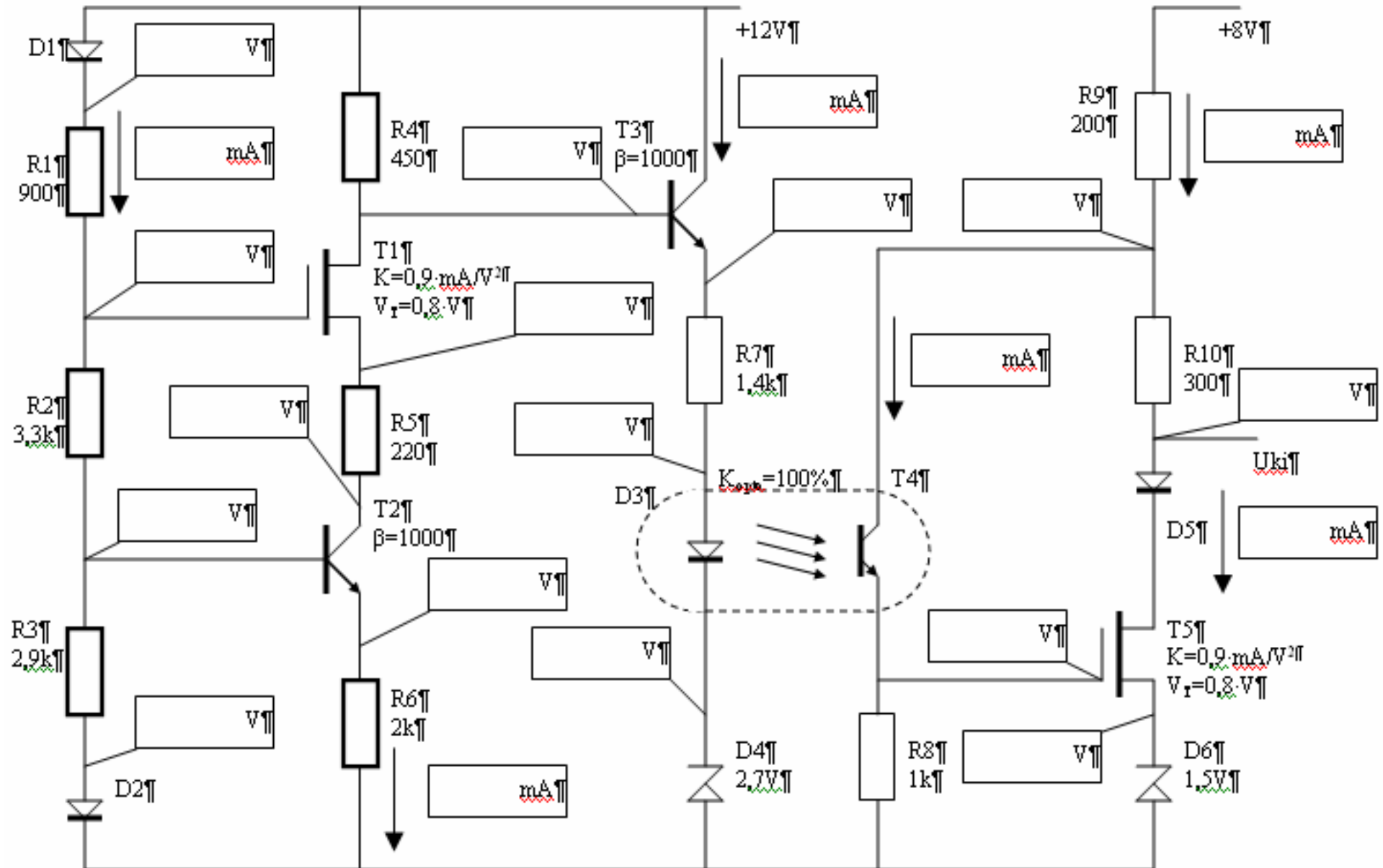
- SWAP Rd ; exchange Rd(7..4) Rd(3..0)
- TRAP #n ; trap to subroutine
- XCHB Rd ; exchange with B
- LDSP ;Load stack pointer
- STSP ;Store stack pointer



Mintapéldák

Határozza meg az alábbi áramkör munkaponti áram és feszültség adatait!

**A1**



Határozza meg az alábbi áramkör munkaponti áram- és feszültség adatait!

**B1**

