Digitális technika

XVII.
Mikroprocesszor
Mikrovezérlő

17.1. Mikroprocesszor, mikrovezérlő

Microprocessor (µP)

- Egy integrált áramkörben (IC-ben) megvalósított CPU egység, a számítógép központi egysége
- Tehát tartalmaz: vezérlőegységet és műveletvégző egységet
- µP = CU + ALU egy IC-ben
- Egy mikroprocesszorhoz csak memóriákat és periféria illesztőket, perifériákat kell hozzá csatlakoztatni, hogy egy komplett számítógépet kapjunk
- Általában nagyon sok kivezetése (lába) van → a három sín vezetékei miatt

17.1. Mikroprocesszor, mikrovezérlő

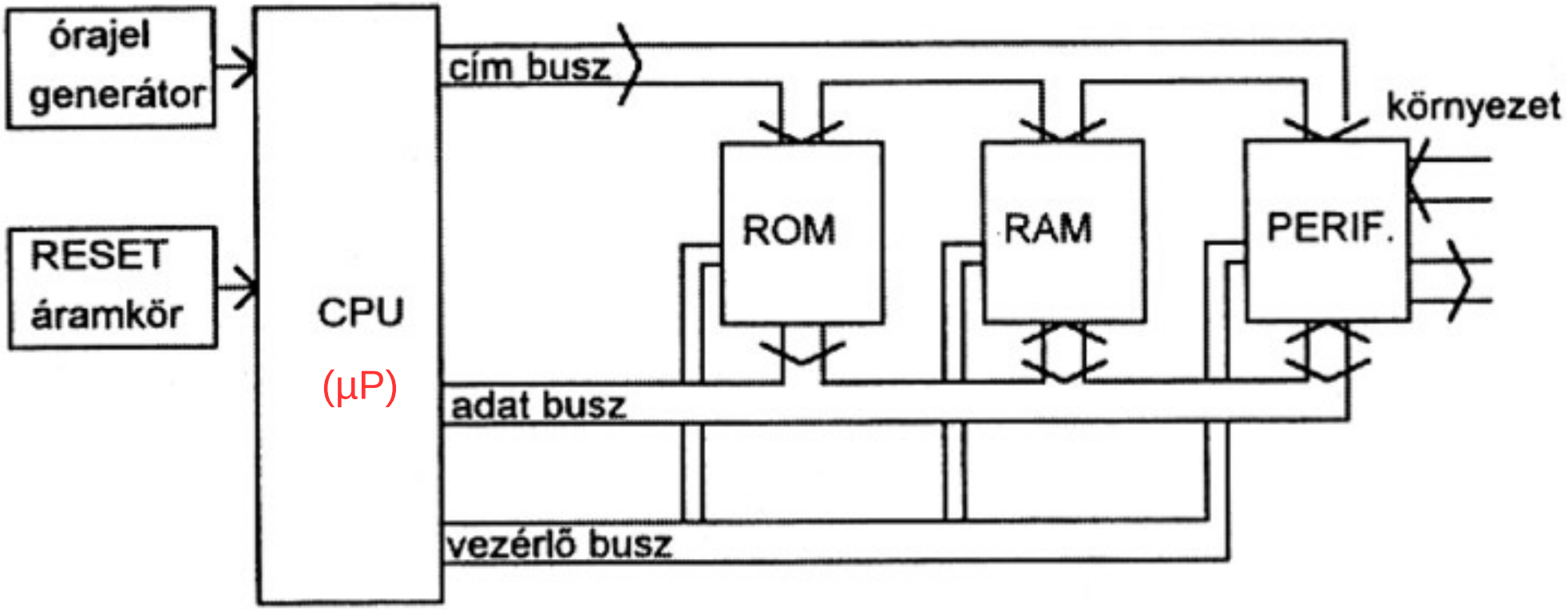
Microcontroller (µC)

- komplett kis számítógép egy integrált áramkörben (IC-ben)
- Tehát tartalmaz: mikroprocesszort, többféle memóriát, és különféle perifériákat, kiegészítő áramköröket
- µC = CU + ALU + memory + I/O egy IC-ben

  → egytokos mikroszámítógép
- Több gyártó cég is van: Atmel, Microchip, Texas, Intel, Analog Devices, ….
- Talán a két legelterjedtebb mikrovezérlő család, az AVR-ek (Atmel) és a PIC-ek (Microchip)
- Mivel számítógépekről van szó ---> programozni kell őket az adott feladat elvégzésére

17.2. Mikroprocesszorok felépítése

Mikroszámítógépek felépítése

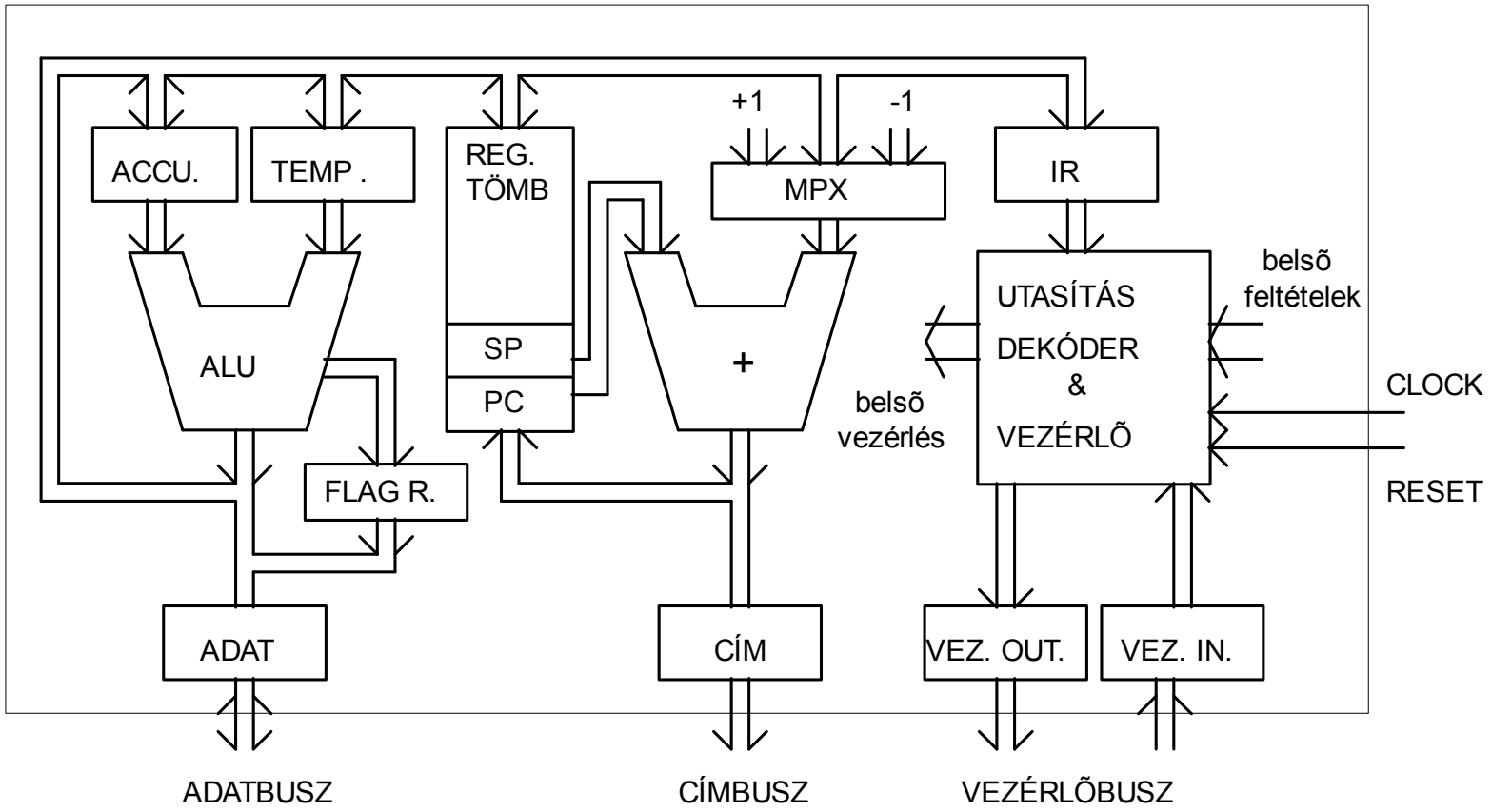


A mikroprocesszor (µP)

- egy CPU funkciót megvalósító áramkör (egy IC-ben)

17.2. Mikroprocesszorok felépítése

A mikroprocesszorok felépítése:



A mikroprocesszorok tartalmaz tehát:

- vezérlő egységet (CU), műveletvégző egységet (ALU),
- belső adatbuszt, címbuszt, vezérlő jeleket,
- általános és speciális célú regisztereket (ACC, PC, IR, SP, FLAG)
- külső címbusz, adatbusz, vezérlőbusz illesztő, meghajtó áramköröket

17.2. Mikroprocesszorok felépítése

A mikroprocesszorok működésének vezérlése:
- a memóriában tárolt program utasításai vezérlik a mikroprocesszort

A mikroprocesszorok legfontosabb képességei:
- az utasítások és adatok megcímzése
- a beolvasott utasítás dekódolása (értelmezése) és végrehajtása → az egységek vezérlése az utasításnak megfelelően (CU – vezérlő egység)
- műveletek elvégzése (ALU) → aritmetikai, logikai
- buszokon zajló adatforgalom vezérlése
- adatok mozgatása memória és regiszterek, illetve regiszterek között
- feltételes ugró utasítások végrehajtása (feltételes elágazás)
- megszakítás kezelés (interrupt)
- veremtár (stack) használata
- buszok vezérlésének felfüggesztése → közvetlen adatcsere lehetősége memória és perifériák között (DMA)

A kommunikáció az egységek között:
- Címzés (címbusz vagy címsín) →a processzor kiválasztja, megcímzi a memóriát (rekeszt) vagy perifériát (annak egy regiszterét) amellyel adatot akar cserélni
- Vezérlő busz (vagy vezérlő sín) → a processzor a vezérlő vezetékekkel beállítja a kommunikáció irányát (olvasás/írás), típusát (memória/periféria), és az időzítéseket
- Adatbusz (vagy adatsín) → adatok továbbítása a processzor és a memória vagy a processzor és valamelyik periféria között

17.2. Mikroprocesszorok felépítése

Speciális funkciójú regiszterek

PC (program counter)
- utasítás számláló → a következő végrehajtandó utasítás címét tartalmazza
- normál program végrehajtás esetén → értéke minden utasítás végrehajtása alatt automatikusan 1-el növekszik
- ugró, függvényhívó utasítások esetén → a kívánt ugrási cím töltődik bele

IR (instruction register)
- utasítás regiszter → az aktuális, beolvasott (végrehajtandó) utasítást tárolja → az utasítás értelmezését az utasítás dekóder áramkör végzi

Accumulator
- AC vagy ACC vagy A vagy W (work register → PIC-ek esetén) jelölésekkel
- az ALU műveletek egyik operandusát és az eredményét tárolja (általában)

Flag (F)
- jelzőbiteket tárol → az ALU által végzett műveletek állítják általában ezen biteket

Z (zero) → 1 ha nulla az eredmény S (sign, előjel) → 1 ha negatív az eredmény

C (carry) → 1 ha átvitel volt P (parity) → 1, ha az eredményben páros számú 1-es van

SP (stack pointer)
- veremtár mutató → a verem memória aktuális címét tartalmazza

Index regiszterek
- indexelt címzéshez, nem minden processzor esetén vannak

Bázis regiszterek
- bázis relatív címzéshez, nem minden processzor esetén vannak

17.3. Processzorok speciális üzemmódjai

Megszakítás (interrupt)

- a program normál végrehajtását egy rövid időre felfüggesztjük, megszakítjuk
  → hogy valamilyen nagyon fontos, abban a pillanatban bekövetkező (általában külső) esemény hatását lekezeljük, az eseményre reagáljunk
- az eseményre szükséges válaszlépések elvégzése után folytatódik a normál program végrehajtása (attól a ponttól ahol megszakítottuk)
- általában valamilyen periféria generálja a mikroprocesszor felé → külső megszakítás bemenetük van a processzoroknak !! (akár több is),
  pl. INT, NMI (nem maszkolható → nem tiltható)
- de generálható megszakítás szoftveres úton is

Megszakítás használata

- a megszakítás lekezelése hasonló mint egy függvény használata → ilyenkor is ugrás történik → egy speciális, megszakításkezelő függvény hívódik meg !!
- De a különbség az, hogy megszakításkor automatikusan történik a megszakításkezelő függvény meghívása !!
  Hiszen a megszakításnál pont az a lényeg, hogy nem tudjuk mikor fog megtörténni !! (ilyen szempontból váratlan) → de nagyon gyorsan reagálnunk kell rá
- A feladatunk csak az, hogy a megszakításkezelő függvényben lévő utasításokat megadjuk → mi történjen az esemény bekövetkezte esetén
- A megszakítások általában engedélyezhetők/tilthatók → programozással

17.3. Processzorok speciális üzemmódjai

DMA (direct memory acces)

- közvetlen, gyors adatcsere lehetősége memória és perifériák között
  → a processzor kihagyásával !
- a processzor ilyenkor a buszok vezérlését átadja egy speciális hardvernek
  → egy DMA vezérlő áramkörnek
- DMA átvitel alatt a processzor tehermentesítődik
- külön vezérlő bemenet és kimenet szükséges ehhez a processzorban
  → pl. HOLD és HLDA vagy BUSRQ és BUSACK

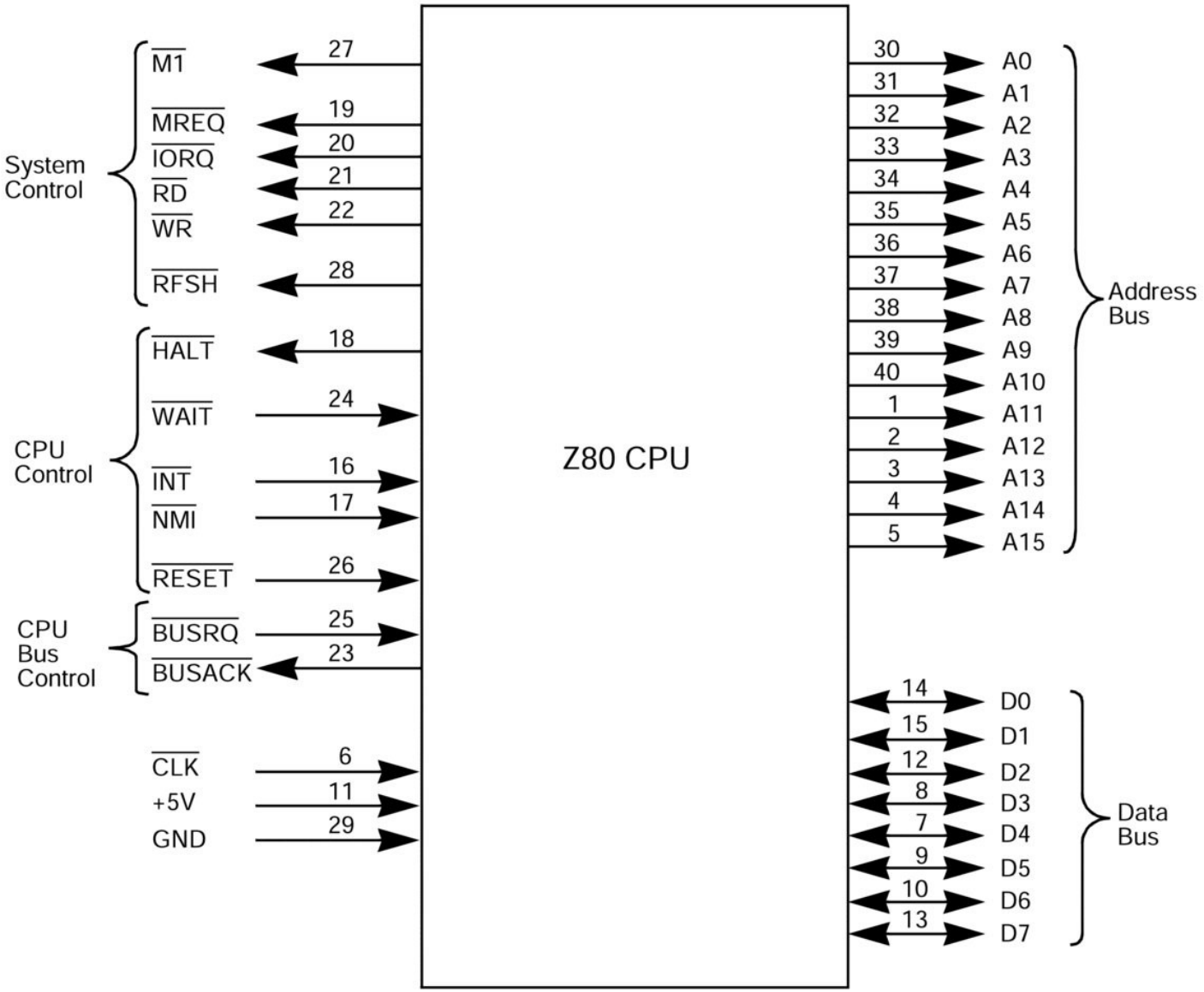
Leállítás, várakozás

- a HALT utasítás hatására a mikroprocesszor (megszakításra) várakozó állapotba kerül → nem csinál semmit ! (NOP utasítások)
- ebből az állapotból egy Reset jel vagy egy megszakítás tudja kimozdítani

Reset

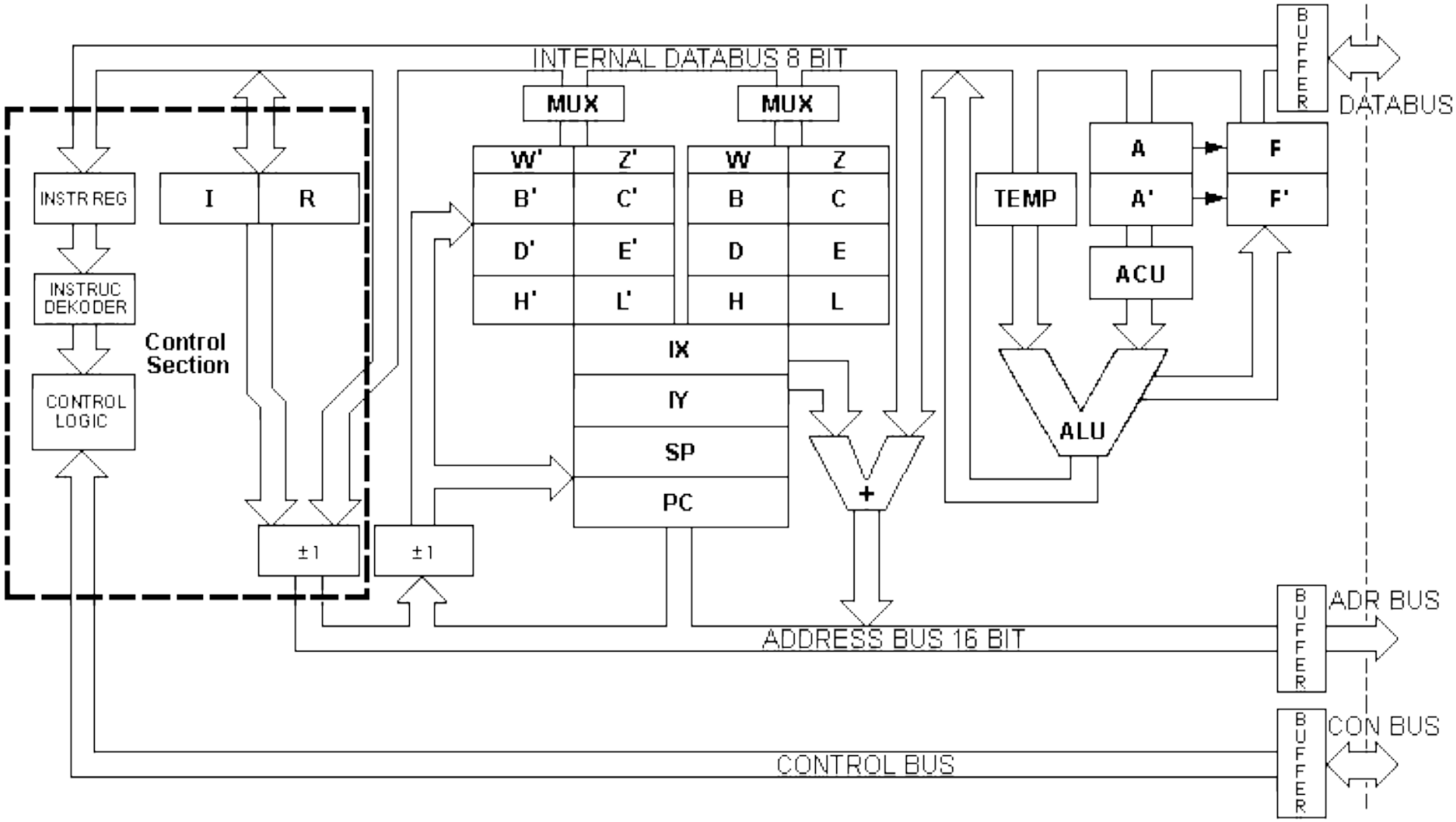
- a RESET vezérlő bemenet hatására a mikroprocesszor alaphelyzetbe kerül → a regiszterek (PC, IR,) valamilyen alap értékeket vesznek fel
- a programszámláló regiszter (PC) is alapértéket vesz fel (általában ez a 0)
  → a memóriában lévő program végrehajtása elkezdődik újra elölről
- megszakítások tiltva

17.4. Z80 mikroprocesszor



17.4. Z80 mikroprocesszor

Zilog Z80 belső felépítése



17.4. Z80 mikroprocesszor

Fontosabb jellemzői

- 8 bites adatokkal dolgozik → adatsín 8 bites (8 vezeték)
- 16 cím vezeték → 16 bites címek → 2^{16} memória rekesz megcímezhető (64kByte)
- tápfeszültség 5V

Regiszterei

IR - utasítás regiszter
I - címzés megszakításhoz
R - dinamikus RAM-ok tárfrissítő címeihez

16 bitesek
PC - utasítás számláló
SP - veremtár mutató
IX, IY - index regiszterek

8 bitesek duplázva vannak !
A, A' - akkumulátor
F, F' - Flag regiszter
B, C, B', C' - általános célú
D, E, D', E' - általános célú
H, L, H', L' - általános célú

Vezérlő kimenetek

aktív 0 szintűek

- $\overline{\text{MREQ}}$ → memória művelet
- $\overline{\text{IORQ}}$ → periféria művelet
- $\overline{\text{RD}}$ → olvasás művelet
- $\overline{\text{WR}}$ → írás művelet
- $\overline{\text{M1}}$ → Fetch gépi ciklus jelzése
- $\overline{\text{RFSH}}$ → tár frissítő cím a sinen
- $\overline{\text{BUSACK}}$ → DMA kérés elfogadása
- $\overline{\text{HALT}}$ → Halt állapot jelzése

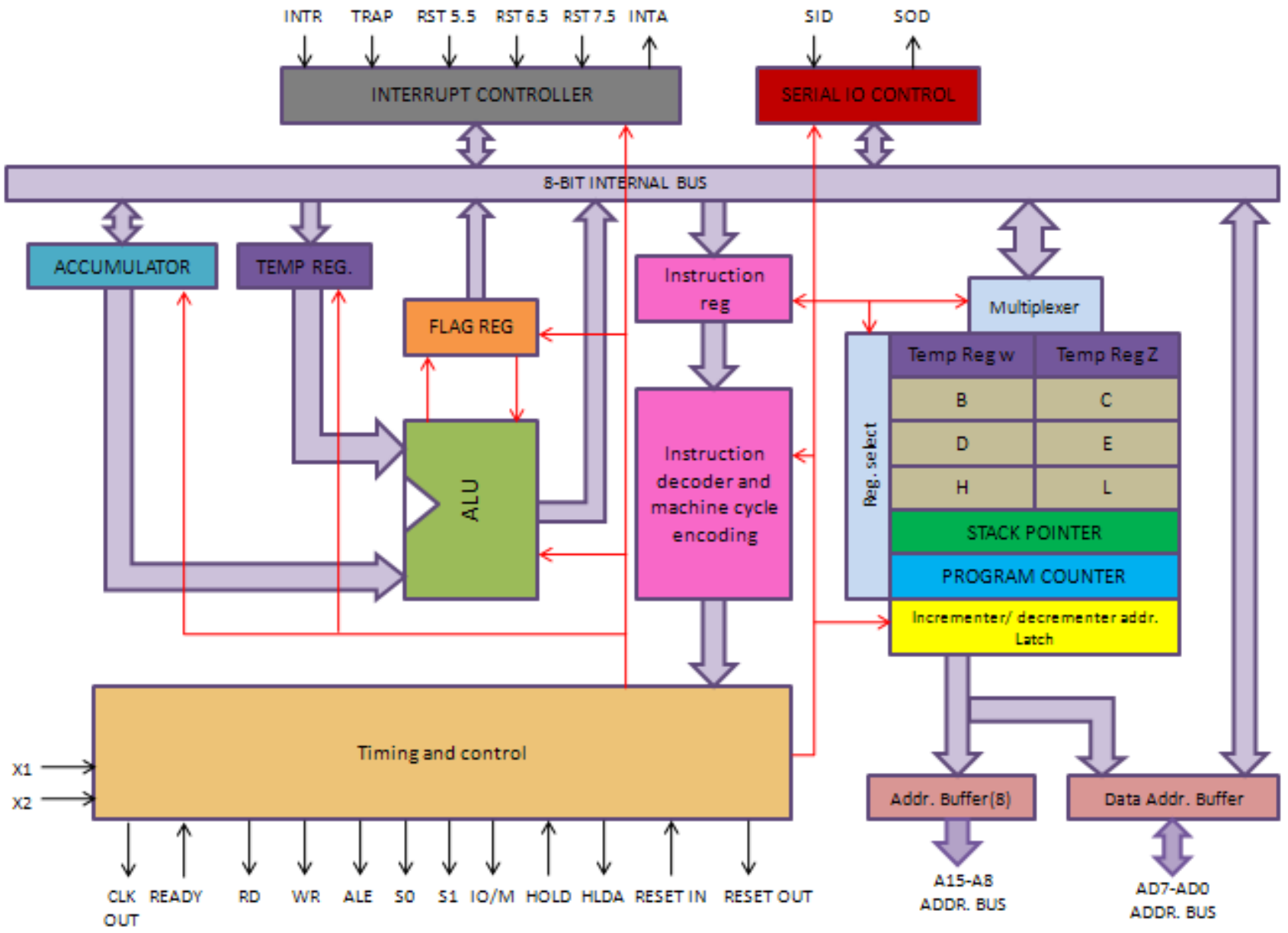
Vezérlő bemenetek

aktív 0 szintűek

- $\overline{\text{CLK}}$ → órajel
- $\overline{\text{RESET}}$ → Reset (újra indítás)
- $\overline{\text{INT}}$ → megszakítás kérés (tiltható)
- $\overline{\text{NMI}}$ → megszakítás kérés (nem tiltható)
- $\overline{\text{WAIT}}$ → várakozási ciklusok kérése
- $\overline{\text{BUSRQ}}$ → DMA kérés

17.5. i8085 mikroprocesszor

Intel 8085



17.5. i8085 mikroprocesszor

Fontosabb jellemzői

- 8 bites adatokkal dolgozik → adatsín 8 bites (8 vezeték)
- 16 cím vezeték → 16 bites címek → 2^{16} memória rekesz megcímezhető (64kByte)
- tápfeszültség 5V

a címsín alsó 8 bitje és a 8 bites adatsín közös lábakon !!

Regiszterei

IR - utasítás reg.
F - flag regiszter

16 bitesek
PC - utasítás számláló
SP - veremtár mutató

8 bitesek
ACC - akkumulátor
B, C - általános célú regiszterek
D, E - általános célú regiszterek
H, L - általános célú regiszterek

Vezérlő kimenetek

- $\overline{IO/M}$ → port (1)/ memória (0) művelet
- $\overline{RD}$ → olvasás művelet
- $\overline{WR}$ → írás művelet
- ALE → AD_7-AD_0 lábakon cím !
- HLDA → DMA kérés elfogadása
- RESETOUT → Reset ki (Resetin-re)
- $\overline{INTA}$ → megszakítás kérés elfogadása
- S_0,S_1 → belső művelet típusa
- CLK → órajel kimenet
- SOD → soros adat kimenet

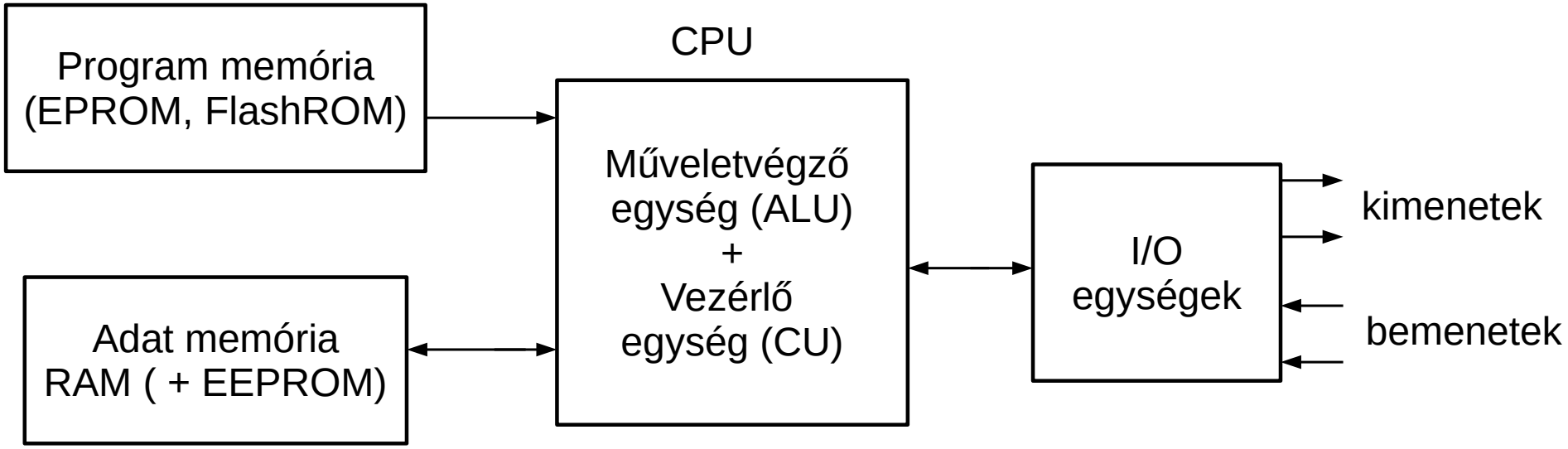
Vezérlő bemenetek

- X1, X2 → kvarc órajel generátorhoz
- $\overline{RESETIN}$ → Reset (újra indítás)
- HOLD → DMA kérés
- READY → várakozási ciklusok kérése
- INTR → megszakítás kérés
- RST 5.5 6.5 7.5 → megszakítás kérés
- TRAP → megszakítás kérés (nem tiltható)
- SID → soros adat bemenet

17.6. Mikrovezérlő

Mikrovezérlők felépítése

Nem teljesen a hagyományos számítógép felépítést követik →
az adat és program memória külön van választva → Harvard architektúra



Jellemző perifériák: digitális bemenetek, digitális kimenetek, analóg bemenetek, időzítők, számlálók, komparátorok, kommunikációs portok (RS232, SPI, I^2C, USB)

17.7. PIC16F887 mikrovezérlő

- PIC → A Microchip cég gyártja ezen mikrovezérlőket
- PIC rövidítés,----> Programmable Interface Controller

  eredetileg: Peripheral Interface Controller ?

PIC-ek csoportosítása

* utasításhossz alapján lehet: 12,14,16,24 vagy 32 bites

* adathossz alapján lehet: 8,16 vagy 32 bites

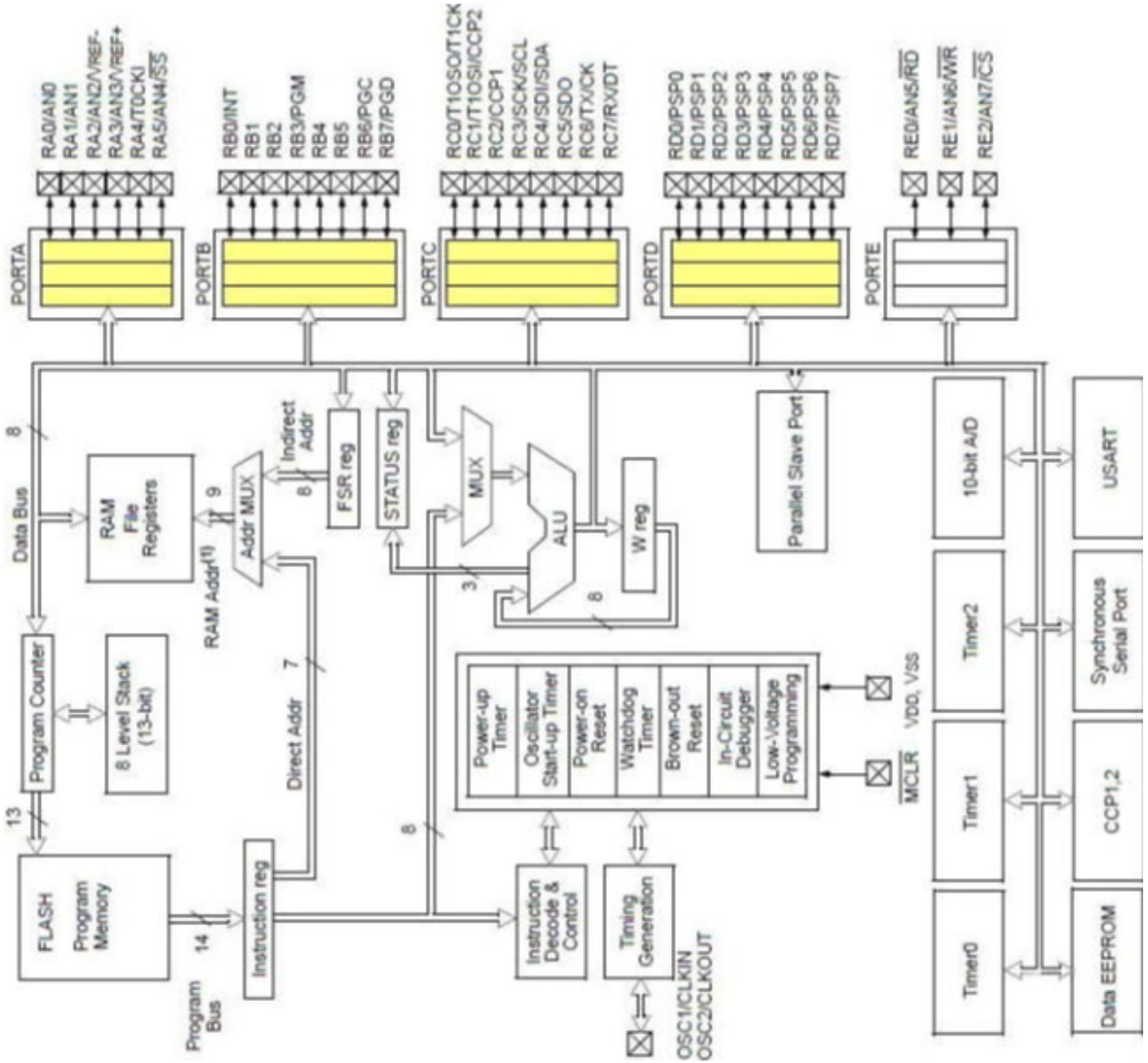
	12bit	14bit	16bit	24bit	32bit
8bit	PIC10 PIC12	PIC14 PIC16	PIC18		
16bit				PIC24 dsPIC	
32bit					PIC32

pl.
PIC16F887,
PIC18F2550

PIC16xyz → közepes teljesítményű, 8 bites mikrovezérlők

17.7. PIC16F887 mikrovezérlő

PIC16F887



17.7. PIC16F887 mikrovezérlő

- Többféle tokozással készül, DIP → 40 kivezetés
- Tápfeszültség: 2 – 5,5V között (V_{DD} láb(11,32) → + V_{SS} láb(12,31) → -)
- Órajel: 0 – 20 MHz között, külső oszcillátor OSC1(13), OSC2(14) lábakra
  de tartalmaz belső RC oszcillátort (31kHz - 8 MHz)

Utasítás végrehajtás
- RISC processzor, kevés számú, egyszerű utasítás
- egy gépi ciklusa (belső ciklus) 4 órajel ciklus alatt játszódik le
- az utasítások 14 bitesek, és 8 bites adatokkal dolgozik

Memóriák
- program memória, FlashROM 8kszó (8k x 14 bit)
- adatmemória, RAM 368 byte + regiszterek (hardver, perifériák vezérlésére)
  EEPROM 256 byte, 'háttértár' → adatok stabil tárolására

Üzemmódjai
- normál program végrehajtás → $\overline{\text{MCLR}}$/VPP (1) lábra tápfeszültség
- RESET (újra indítás) → $\overline{\text{MCLR}}$/VPP (1) lábra 0 szint
- programozás (ICSPDAT-40, ICSPCLK-39 lábakon) → $\overline{\text{MCLR}}$/VPP (1) lábra 12V

17.7. PIC16F887 mikrovezérlő

A legtöbb kivezetésnek/lábnak több funkciója is van → a kívánt funkciót programozással, a megfelelő regiszterek (SFR) bitjeinek állításával lehet kiválasztani

Perifériák
- 35 (36) digitális bemenet/kimenet, (programból kell állítani, hogy be vagy ki)
- 14 analóg bemenet (AN0, AN1, AN2, … AN13)
- 3 időzítő/számláló (timer)
- kommunikációs portok, USART (RS232,RS485), MSSP (SPI, I^2C) ...

A digitális bemenetek/kimenetek 8-as csoportokba vannak szervezve, és így vezérlő regiszterekhez rendelve
- RA0, RA1,RA2, ...RA7 → PORTA, TRISA regiszterek
- RB0, RB1,RB2, ...RB7 → PORTB, TRISB regiszterek
- RC0, RC1,RC2, ...RC7 → PORTC, TRISC regiszterek
- RD0, RD1,RD2, ...RD7 → PORTD, TRISD regiszterek
- RE0, RE1,RE2, (RE3) → PORTE, TRISE regiszterek

-TRISx regiszter bitjei állítják be az irányokat, ha 0 → kimenet, ha 1 → bemenet
- PORTx regiszter bitjein keresztül pedig a hozzá rendelt lábakra lehet írni, vagy be lehet olvasni a láb értékét (attól függően, hogy éppen be- vagy kimenetnek van beállítva

17.8. Mikrovezérlők speciális áramkörei

Analóg-digitális konverter

A mikrovezérlőben egy analóg-digitális átalakító van mint speciális periféria.
Segítségével a mikrovezérlő analóg jeleket tud fogadni a külvilág felől, és azt digitálisan fel tudja dolgozni.
A PIC16F887 esetében 10 bites az AD konverter, és 14 db analóg csatorna van (ANSx)
→ 14 láb lehet analóg bemenet

Timer

PIC16F887 mikrovezérlőben 3 Timer áramkör van:
Timer0 – 8 bites Timer1 – 16 bites Timer2 – 8 bites
Mindegyik timerhez több regiszter is tartozik → amelyikben a számlálás folyik
+ a beállító regiszterek (a számlálás forrása, előosztó beállítása, …)
Ezek igazából számláló áramkörök:
- számolhatnak valamelyik külső lábon érkező impulzusokat (fel- vagy lefutó élt)
- számolhatnak egy meghatározott frekvenciájú belső órajel impulzusait → a számláló értékei adott időtartamnak felelnek meg → időzítésre használható

Használatuk: amikor a számlálást végző regiszter végállapotból újra a 0 állapotba fordul
→ ez megszakítást okoz → jelzi a megadott idő leteltét
(vagy megadott számú impulzus beérkezését)
Általában programozható előosztó is tartozik hozzájuk (esetleg még utóosztó is), amellyel a számlálási frekvencia csökkenthető

17.8. Mikrovezérlők speciális áramkörei

Órajel hardver

A PIC mikrovezérlők általában többféleképpen kaphatnak órajelet
- külső órajel (EC- external clock) az OSC1 lábon
- beállítható frekvenciájú belső órajel generátor
  külső kvarc (vagy kerámia rezonátor) rákapcsolásával (OSC1, OSC2 lábakon),
  vagy külső R-C elem rákapcsolásával (OSC1 lábon)
- belső R-C oszcillátor, (pl. 31,25kHz-8MHz 8 választható értékkel)

Watchdog timer (WDT)

- biztonsági időzítő áramkör → üzembiztos működés biztosítása instabil üzemi körülmények esetén
- egy külön, belső RC oszcillátor és számláló áramkör
- időzítése (idő kifutása) ~ 15-30ms → ekkor megszakítást generál (ha engedélyezve van)
- az időzítés növelhető egy 8 bites utóosztóval (Timer0 előosztója)
- a normál programban tehát időnként a WDT számlálóját nullázni kell (CLRWDT), hogy ne okozzon megszakítást
- a program lefagyásakor ez a törlés elmarad → megfelelő idő után megszakítás → Reset

Belső reset áramkör

Meghatározott kezdeti állapotba állítja a mikrovezérlőt → a programot elkezdi újra elölről végrehajtani,...
Többféle esemény is kiválthatja a reset folyamat elindulását

17.8. Mikrovezérlők speciális áramkörei

CCP modulok

1. Capture (kiolvasás)
- számláló helyzet (Timer1) elmentése (regiszterekbe) külső esemény hatására (CCPx láb)
  → majd megszakítás

Felhasználása:
- váratlan külső eseményre gyors reagálás, az esemény idejének ismeretében !
- két külső esemény között eltelt idő mérése

2. Compare (összehasonlítás)
- a léptetett Timer1 értékének összehasonlítása egy előre beállított értékkel → egyezés esetén → CCPx lábon logikai szint váltás, vagy megszakítás kérés

3. PWM (impulzus szélesség modulált jelgenerátor)
- állandó frekvenciájú (periódusidejű) nagyfrekvenciás jel (néhányszor 10kHz), amelynek a kitöltési tényezőjét változtatva (impulzus szélesség moduláció) →változik a kimenőjel átlag feszültsége → vezérelni, szabályozni tudjuk egy áramkör, fogyasztó teljesítményét

Parallel Slave Port (PSP)

Mikroprocesszoros rendszerbe illesztést lehet megvalósítani alkalmazásával
- PORTD 8 lába (PSP0 – PSP7) → adatbuszra (D0 – D7)
- PORTE 3 lába → $\overline{CS}$ (chip select → címzés), $\overline{RD}$ (olvasás) és $\overline{WR}$ (írás) vezérlő bemenetek

17.8. Mikrovezérlők speciális áramkörei

MSSP kommunikációs port

Master Synchron Serial Port → SPI és I2C kommunikáció

I2C (Inter-Integrated Circuit)
- kétvezetékes soros kommunikációs sín → SCL (órajel), SDA (adat)
- több eszköz → a csatlakoztatott eszközök címezhetőek !
- kétirányú, soros, 8-bites adatforgalom (+ Start, Stop, ACK)

SPI (Serial Peripheral Interface)
- „négy” vezetékes soros kommunikációs sín (4, 3 vagy 5 vezeték)
  SDI (bemenet), SDO (kimenet), SCK (órajel), SS (Slave Select)
- kétirányú szinkron soros kommunikáció (duplex !)
- két eszköz között, master-slave

USART kommunikációs port

univerzális szinkron/aszinkron adó-vevő

RS232C kommunikáció megvalósítása
- aszinkron, 8+1 bites (+ Start, Stop) kommunikáció
- RX (vétel), TX (adás) lábak

Szinkron soros kommunikáció
- DT (adat), CK (órajel) lábak

17.9. Mikrovezérlők speciális üzemmódjai

Megszakítások mikrovezérlőknél

- megszakítást tudnak kiváltani a beépített periféria áramkörök
  - → időzítő (timer) így jelzi hogy vége az időzítésnek
  - → analóg-digitális átalakító (A/D konverter) így jelzi, hogy vége az átalakításnak
  - → kommunikációs interfészek (USART, USB, …)
  - → adat EEPROM
- megszakítást tudunk kiváltani a mikrovezérlő megfelelő lábára adott külső jellel
  - → PIC-ek esetén általában az INT láb szolgálhat külső megszakítás bemenet céljára (RB0/INT)
  - → néhány PORTx (PORTB) láb állapot változása okozhat megszakítást
- egy speciális, megszakításkezelő függvény hívódik meg !!
- a megszakítások hatására állítódnak a mikrovezérlő különböző regisztereiben lévő megszakítás jelző bitek (IF - interrupt flag) → ezek lekérdezésével meg lehet állapítani, hogy milyen megszakítási esemény történt → ez akkor különösen hasznos ha több megszakítás is engedélyezve van
- a megszakítások engedélyezéséhez/tiltásához állítani kell a mikrovezérlő különböző regisztereiben lévő megszakítás engedélyező biteket (IE - interrupt enable)

17.9. Mikrovezérlők speciális üzemmódjai

Reset állapot

Meghatározott kezdeti állapotba megy a mikrovezérlő→ a programot elkezdi újra elölről végrehajtani,...
Többféle esemény is kiválthatja a reset folyamat elindulását:
- külső Reset normál működés közben ($\overline{\text{MCLR}}$ lábra 0 szintet adva)
- külső Reset Sleep állapotban ($\overline{\text{MCLR}}$ lábra 0 szintet adva) → kilép a Sleep állapotból → Reset
- Watchdog Timer reset normál működés közben
- Power on reset, POR → bekapcsolási reset → tápfeszültség megjelenése után egy kis késleltetéssel indul, hogy stabil tápfeszültség és órajel esetén induljon el ténylegesen a program végrehajtása
- Brown out reset, BOR → tápfeszültség lecsökkenése esetén (egy bizonyos szint alá, pl. feszültség ingadozás esetén)

Sleep üzemmód

Készenléti (alvás) üzemmód (SLEEP utasítás) → minimális fogyasztás
- I/O kivezetések tartják logikai állapotukat
- oszcillátor leáll

Kilépés Sleep üzemmódból:
- Reset hatására
- WDT hatására
- Megszakítás hatására

17.10. PIC mikrovezérlők, memória szervezés

Program memória (FlashROM vagy EPROM) felosztása

Mikrovezérlőknél a program memória alapvetően két részből áll
- a nagyobb része a felhasználói program memória → ide töltődik be a program
  a PIC16F887 esetében ez 8kszó kapacitású → 8x1024 x14 bit
  ennek a címtartomány 0000h – 1FFFh
- egy kis méretű konfigurációs memória → ez különféle azonosítókat, beállításokat tárol
  mérete általában csak néhány byte, de mérete és tartalma mikrovezérlő típustól függ
  a PIC16F887 esetében ennek a címtartománya elvileg 2000h – 3FFFh,
  de gyakorlatilag csak 2000h – 2009h van használatban →
  10 szó kapacitású → 10 x14 bit

A konfigurációs memória (PIC16F887)

Több részből áll:
- azonosító mező (2000h – 2003h) → azonosító, ellenőrző összeg, …
- típusazonosító (2006h)
- konfigurációs szó (szavak) → konfigurációs bitek, 3 db regiszterbe rendezve
  CONFIG1 (2007h), CONFIG2 (2008h), CONFIG3 (2009h)

A konfigurációs biteket programból nem tudjuk módosítani !! → a program beírásakor lehet ezeket is beállítani → vagy a programozó szoftverben tudjuk őket módosítani,
→ vagy a program fejlesztő környezetben tudjuk őket beállítani →
fordításkor bekerülnek a hexa állományba

17.10. PIC mikrovezérlők, memória szervezés

Adat memória (RAM) felosztása

Mikrovezérlőknél az adat memória szintén két részből áll feladat szempontjából (elhelyezkedés, címzés szempontjából nincsenek szét választva !!)

1. egyik része a felhasználói adat memória (normál RAM)
   - saját programunk adatainak tárolására (változók)
   - ezt a részt nevezik általános adatregisztereknek (GPR - General Purpose Registers)

2. a másik része vezérlő funkciókat ellátó memória
   - ezen keresztül tudjuk a mikrovezérlő hardver elemeit, perifériákat, speciális áramköröket közvetlenül beállítani, vezérelni, adatokat küldeni számukra ill. adatokat beolvasni tőlük.
   - ezt a részt nevezik hardver vezérlő regisztereknek (SFR - Special Function Registers)
   - ezen terület bitjei Reset hatására meghatározott alapértékeket vesznek fel !

PIC16F887 esetében a teljes címezhető RAM 512 byte kapacitású, ebből
368 byte a normál RAM (GPR) és 90 byte a hardver vezérlő regiszterek (SFR)
a maradék vagy speciális célú, vagy nem használt

Memória lapok (memory banks)

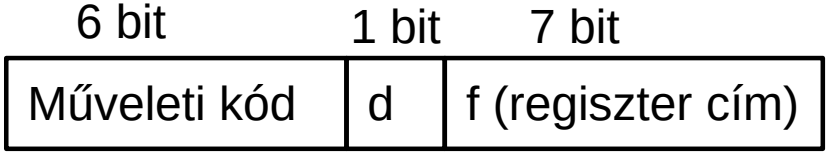
A 14 bites utasításokat használó PIC-ek esetén a memória címzésére 7 bit használható → egyszerre csak 128 cím van → a nagyobb RAM-ok felosztásra kerülnek 128 bájtos lapokra → egyszerre csak egy lapot használhatunk, a STATUS regiszter tárolja az aktuális lap számát (a tényleges címzés a segítségével történik) → lap váltása → lapozás
PIC16F887 esetében négy memória lap van → Bank0, Bank1, Bank2, Bank3
Minden memória lapon vegyesen vannak SFR és GPR regiszterek

17.11. PIC16F887 utasításai

Utasítás végrehajtás

- RISC processzor, (csökkentett utasításkészletű) csak 35 utasítás
- egy gépi ciklusa (belső ciklus) 4 órajel ciklus alatt játszódik le
- az utasítások nagy része 1 gépi ciklus alatt végrehajtódik
- az utasítások 14 bitesek, és 8 bites adatokkal dolgozik

Byte-orientált utasítások



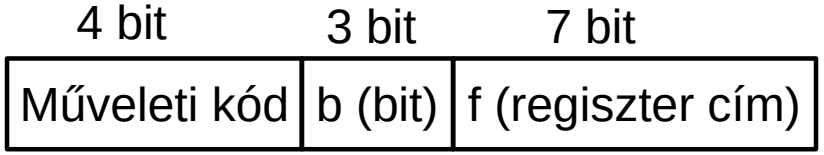
d (destination) → eredmény hová kerüljön
0 → W regiszterbe
1 → fájl regiszterbe (memóriába)

file register → memória rekesz

pl. ADDWF f,d W és egy fájl regiszter összeadása
(000111dfffffff) ha d=0 (W)+(f) → (W)
ha d=1 (W)+(f) → (f)
SUBWF f,d W kivonása egy fájl regiszterből
(000010dfffffff) ha d=0 (f)-(W) → (W)
ha d=1 (f)-(W) → (f)
MOVF f,d regiszter tartalmának másolása
ha d=0 (f) → (W)
ha d=1 (f) → (f) saját magába vissza !

17.11. PIC16F887 utasításai

Bit-orientált utasítások



b → 3 bites bit cím

file register → memória rekesz

pl. BCF f,b bit törlése egy regiszterben
(0100bbbfffffff)

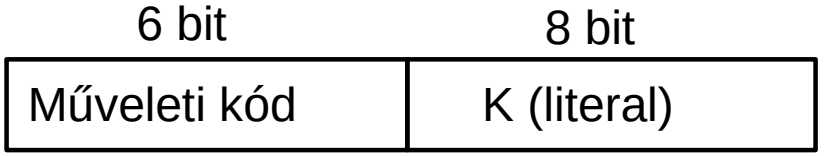
BSF f,b bit 1-be állítása egy regiszterben
(0101bbbfffffff)

BTFSC f,b egy regiszterben bitjének tesztelése
(0110bbbfffffff)

ha értéke 0 → a következő utasítás törölve ! →
helyette NOP !

17.11. PIC16F887 utasításai

Literal (konstanst tartalmazó) utasítások

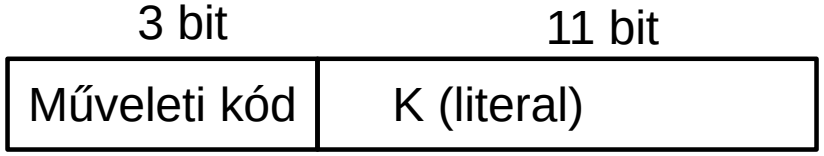


K → 8 bites közvetlen érték

pl. ADDLW K szám (K) hozzáadása W-hez
(11111xKKKKKKKK) (W)+K → (W)

MOVLW K szám (K) betöltése W regiszterbe
(1100xxKKKKKKKK) K → (W)

Ugró utasítások



K → 11 bites közvetlen cím

pl. GOTO K ugrás egy címre (jump)
(101KKKKKKKKKKK) K → (PC alsó 11 bitjébe)
és (PCLATH 4,3 bitje) →(PC felső 2 bitjébe)

CALL K függvény hívás
(100KKKKKKKKKKK) K → (PC alsó 11 bitjébe)
és (PCLATH 4,3 bitje) →(PC felső 2 bitjébe)

17.11. PIC16F887 utasításai

Table 29-1: Midrange Instruction Set

Mnemonic, Operands		Description	Cycles	14-Bit Instruction Word MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	