



**BME**  
Budapesti Műszaki és Gazdaságtudományi Egyetem

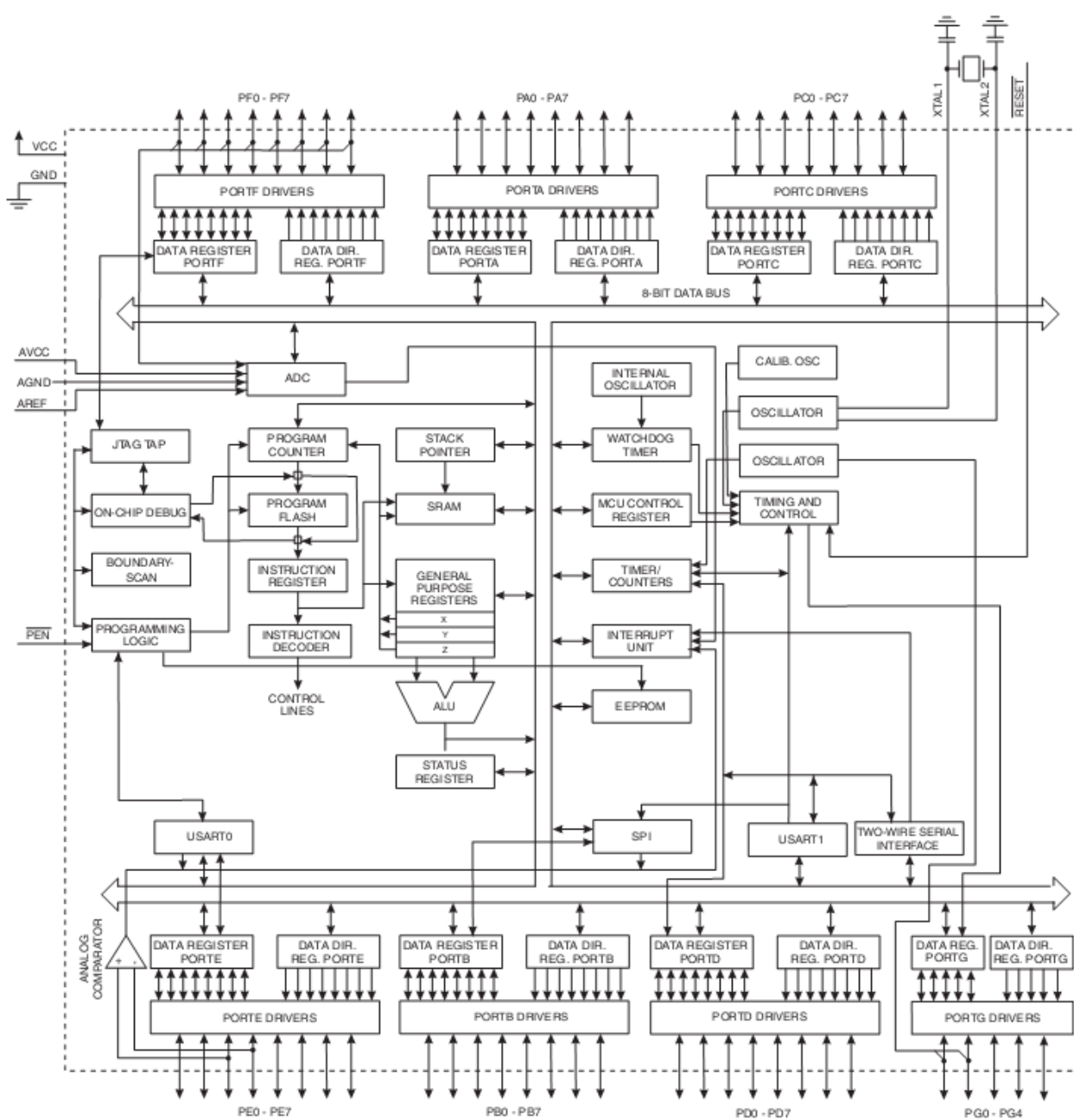
**KAUT**  
Közlekedésautomatikai Tanszék



# Járműfedélzeti rendszerek I.

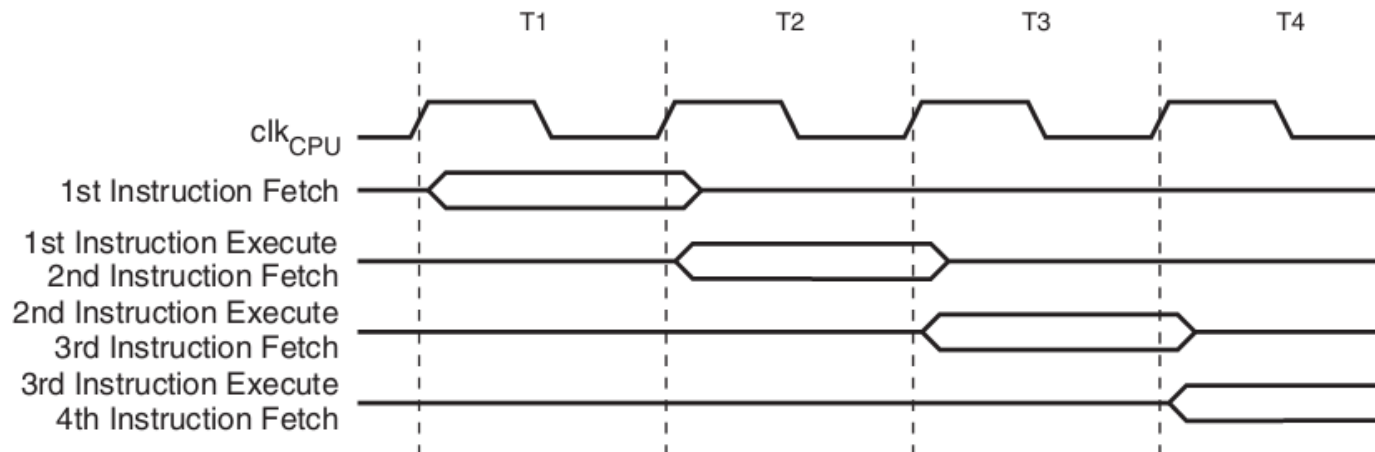
3. előadás

Dr. Bécsi Tamás



# ALU I.

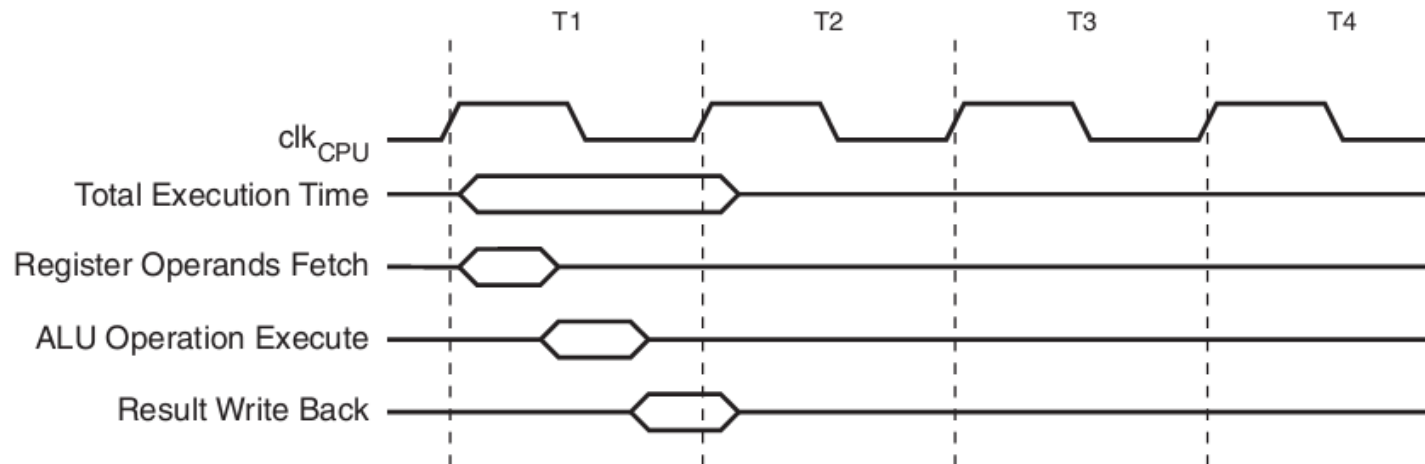
- Utasítások futásának időzítése
  - Egy utasítás futtatása (execute) párhuzamosan fut a következő utasítás betöltésével (fetch). Ez az ún. „single-level pipelining”, így elérhető az 1 MIPS/MHz



# ALU II.

- ALU műveletek időzítése

- Egy ALU által elvégzett művelet egy ciklus alatt lefut.
  - Operandusok betöltése a regiszterekből (Register Operands Fetch)
  - ALU művelet futtatása (ALU Operation Execute)
  - Az eredmények visszairás a regiszterekbe (Result Write Back)



# ALU III.

- Az ALU a műveletvégzések során frissíti a státusz regisztert (**SREG**), amely az alábbi jelzőbiteket (flag)tartalmazza
  - **I – Global Interrupt Enable**: globális megszakítás engedélyezése
  - **T – Bit Copy Storage**: tetszőleges regiszter egy bitjének tárolására szolgál, amely utána elágazó utasítások feltétele lehet
  - **H – Half Carry Flag**: félátvitel jelzőbit, amely az alsó négy bitről a felső négyre történő átvitelt jelzi.
  - **S – Sign Bit**: előjel bit, az N és V bitek kizáró vagy kapcsolata
  - **V – Two's Complement Overflow Flag**: kettes komplement túlcsordulás
  - **N – Negative Flag**: negatív jelző bit, értéke az eredmény legfelső bitje
  - **Z – Zero Flag**: nulla jelzőbit, melynek értéke egy, ha az eredmény nulla
  - **C – Carry Flag**: átvitel jelzőbit

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

# Általános célú regiszterek

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- 32 darab 8-bites regiszter, melyek közvetlenül kapcsolódnak az ALU-hoz, így bármely regisztereken végezhető művelet
- A regiszterek egyenrangúak, két kivétellel
  - LDI (Load Immediate) utasítás csak a felső 16 regiszterre (R16-R31) van értelmezve
  - Az X, Y, Z 16-bites regiszterek megfelelnek az R26-R27, R28-R29, R30-R31 regiszterpároknak

7	0	Addr.
	R0	\$00
	R1	\$01
	R2	\$02
	...	
	R13	\$0D
	R14	\$0E
	R15	\$0F
	R16	\$10
	R17	\$11
	...	
	R26	\$1A
	R27	\$1B
	R28	\$1C
	R29	\$1D
	R30	\$1E
	R31	\$1F

R26	\$1A	X-register Low Byte
R27	\$1B	X-register High Byte
R28	\$1C	Y-register Low Byte
R29	\$1D	Y-register High Byte
R30	\$1E	Z-register Low Byte
R31	\$1F	Z-register High Byte

# Egyéb fontos CPU regiszterek

- **Program Counter (PC):** 16-bites programszámláló, amely az aktuális utasítás helyére mutat a programmemóriában
- **Stack Pointer (SP):** 16-bites veremtár mutató, mely két 8-bites regiszterrel (SPH, SPL) van megvalósítva. A veremtárnak az SRAM-ban (>\$0060 memóriacím) kell helyet lefoglalni a felhasználói programban. A mutató értéket a PUSH utasítás eggyel csökkenti, míg a POP növeli. A szubrutin és megszakítás hívások esetén a PC értéke itt tárolódik, és innen töltődik vissza a RET vagy RETI hatására.
- **RAM Page Select Z (RAMPZ):** Csak a legkisebb helyiértékű bit használatos belőle. Meghatározza, hogy az SRAM alsó, vagy felső 64 kB-ját címzi-e a Z regiszter, amennyiben az adott AVR típus ezt támogatja. Az ATmega128 csak max. 64 kB SRAM-ot támogat, ezért itt a 128 kB programmemória megfelelő részeit lehet kiválasztani vele
- **MCU Control Register (MCUCR):** külső memória kezelés, alvó módok beállítása, IT vektor áthelyezés
- **MCU Control and Status Register (MCUCSR):** reset jelzőbitek, JTAG engedélyezés
- **Oscillator Calibration Value (OSCCAL):** belső RC oszcillátor kalibrációs érték
- **XTAL Divide Control Register (XDIV):** az órajel leosztására használható
- **External Memory Control Register A & B (XMCRA, B):** külső memória vezérlés

# Fuse bitek

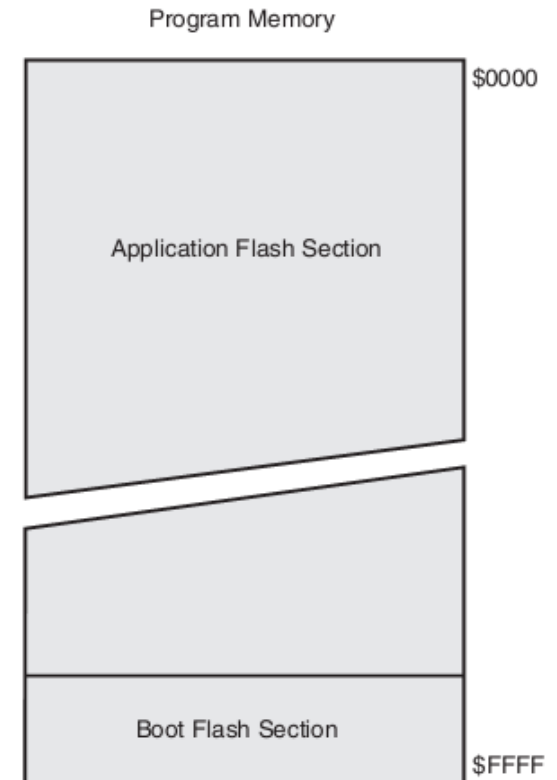
- A Flash programmemória konfigurációs bitjeit ún. fuse (biztosíték) biteknek hívjuk. Programozott állapotban 0 értéket vesznek fel.
  - ATmega103 kompatibilitási mód
  - Watchdog engedélyezés
  - Debug, JTAG, SPI programozási módok engedélyezése
  - Oszcillátor és indulási idők konfigurálása
  - Tápfeszültség figyelés (Brown out detector) engedélyezése
  - Reset vector, boot memória terület beállítása
  - EEPROM védelem beállítása



# Programmemória

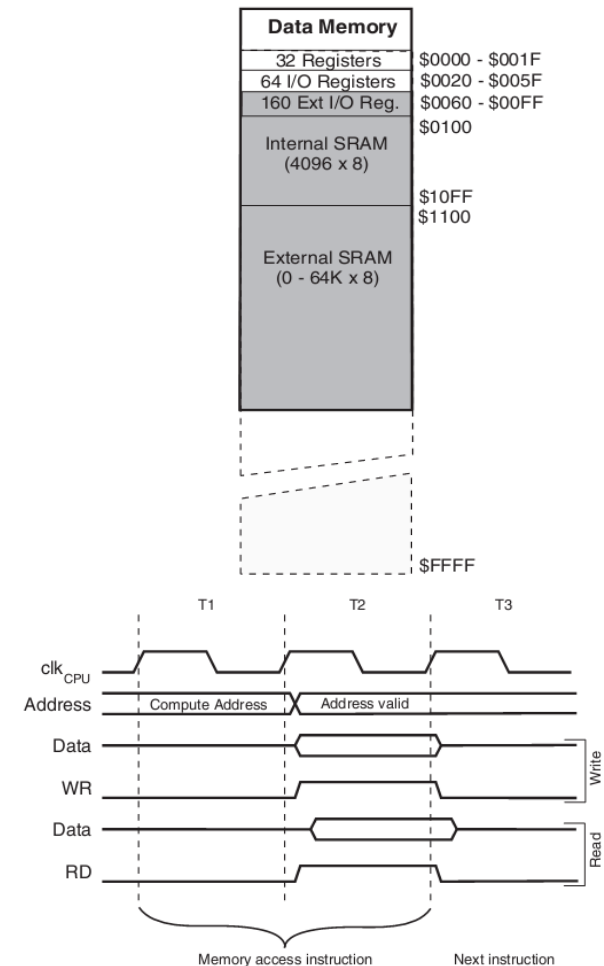
Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- 128 kB Flash alapú memória
- 64k x 16 bit szervezésű (így elég a 16 bites programszámláló)
- 10 000 írási/törlési ciklus élettartam
- Két részből áll:
  - Alkalmazás
  - Boot (levédhető bootloader készíthető)
- Programozási módok:
  - SPI, JTAG, Parallel Programming
  - Self-Programming



# Adatmemória

- Az adatmemória alsó címein közvetlenül elérhető a 32 általános célú regiszter
  - Fizikailag nem az adatmemóriában helyezkednek el
- A következő terület tartalmazza a 64 I/O regisztert, melyek elérhetők direkt I/O címezéssel, vagy általános címezési módokkal
  - I/O direkt címezés (IN/OUT utasítás, 6 bites címtartomány) esetén \$00-val kezdődnek a címek!
  - Az alsó 32 I/O regiszter (5 bites címtartomány) bitenként is elérhető a megfelelő utasításokkal
- Ezt követi a 160 Extended I/O regiszter
  - Ez a terület I/O direkt címezéssel nem érhető el, csak általános címezési módokkal!
- E felett helyezkedik el a belső SRAM, valamint az (opcionális) külső SRAM
  - A külső SRAM használatát külön engedélyezni kell
  - A külső SRAM címei folytonosak, a címezési módok pedig megegyeznek a belső SRAM-mal
  - A belső SRAM elérési ideje két órajel ciklus bájtonként, míg a külsőé három órajel ciklus



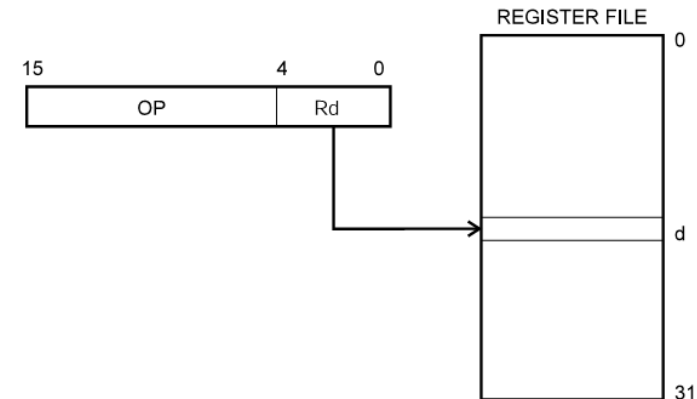
# Címzési módok I.

- Azonnali adatmegadás (Immediate Addressing Data)
- Bit címzés (Bit Addressing)
- Egy regiszter közvetlen címzése (Register Direct, Single Register)
- Két regiszter közvetlen címzése (Register Direct, Two Registers)
- I/O közvetlen címzése (I/O Direct Addressing)
- Adatmemória közvetlen címzés (Direct Data Addressing)
- Adatmemória közvetett címzése (Data Indirect Addressing)
- Adatmemória közvetett címzése eltolással (Data Indirect with Displacement)
- Adat közvetett címzése előzetes csökkentéssel (Data Indirect Addressing with Pre-Decrement)
- Adat közvetett címzése utólagos növeléssel (Data Indirect Addressing with Post-Increment)
- Programmemória konstans címzése (Program Memory Constant Addressing)
- Programmemória konstans címzése utólagos növeléssel (Program Memory Constant Addressing with Post-Increment)
- Programmemória közvetlen címzése (Direct Program Memory Addressing)
- Programmemória közvetett címzése (Indirect Program Memory Addressing)
- Programmemória relatív címzése (Relative Program Memory Addressing)

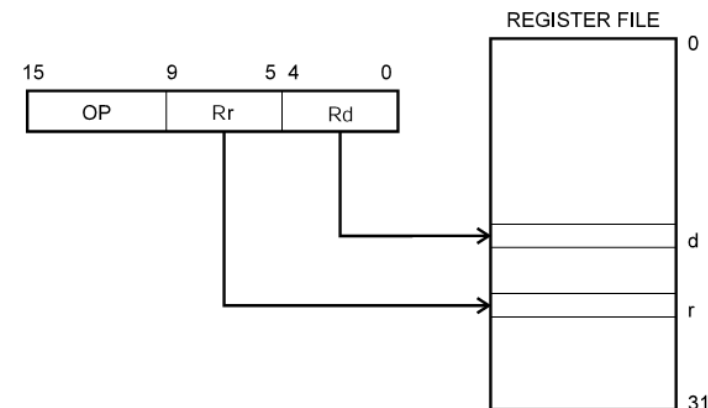
# Címzési módok II.

- Azonnali adatmegadás
  - Az operandus egy 8 bites szám (pl.: LDI R16, 0x55)
- Bit címzés
  - Az operandus az utasítás által meghatározott adatbájt adott sorszámú bitje (pl.: BCLR 7)
- Egy regiszter közvetlen címzése
  - Az operandus és az eredmény is ugyanabban a regiszterben tárolódik (pl.: INC R16)
- Két regiszter közvetlen címzése
  - Az operandus az Rr és Rd regiszterekben, az eredmény az Rd a regiszterben tárolódik (pl.: ADD R17,R16)

Egy regiszter közvetlen címzése



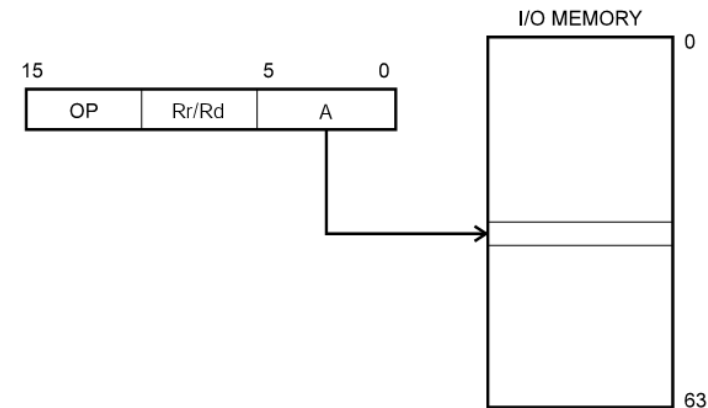
Két regiszter közvetlen címzése



# Címzési módok III.

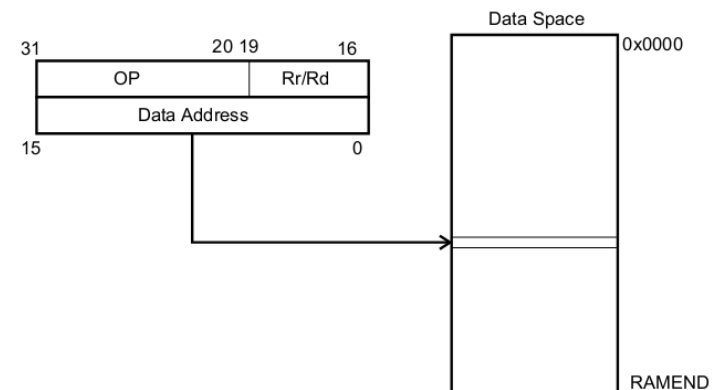
- I/O közvetlen címzése
  - Az alsó 64 I/O regiszter közvetlen címzésére szolgál. Az I/O regiszter 6 bites címét A-val jelöltük, míg a cél/forrás regiszterek Rr/Rd lehetnek (Pl.: IN R16,PINE)

I/O közvetlen címzése



- Adatmemória közvetlen címzése
  - Az adatbájt 16-bites címe az utasítás utolsó két bájtjában helyezkedik el. A forrás/cél regiszter Rr/Rd lehet. (Pl.: LDS R16, 0x0500)

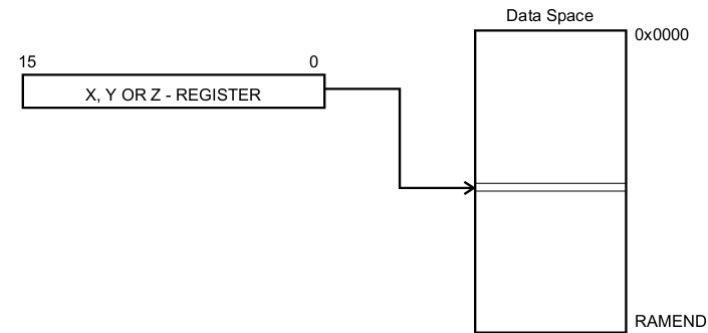
Adatmemória közvetlen címzése



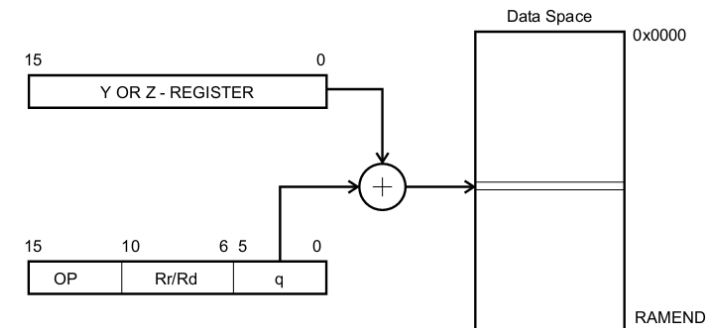
# Címzési módok IV.

- Adatmemória közvetett címzése
  - Az adatbájt 16-bites címe az X, Y, vagy Z regiszterben helyezkedik el. A forrás/cél regiszter Rr/Rd lehet. (Pl.: LD R16, X)
- Adatmemória közvetett címzése eltolással
  - Az adatbájt címe az Y, vagy Z regiszternek és egy 6 bites eltolásnak (q) az összege. A forrás/cél regiszter Rr/Rd lehet. (Pl.: LDD R16, Y+0x10)

Adatmemória közvetett címzése



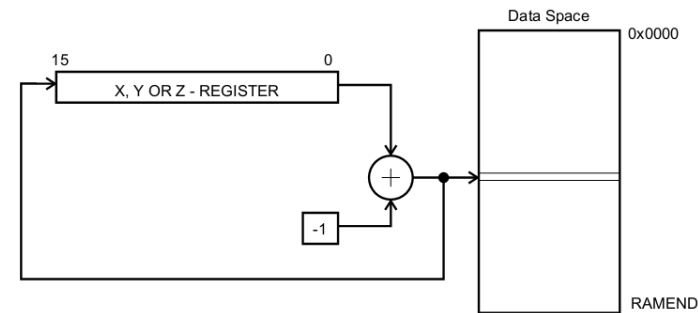
Adatmemória közvetett címzése eltolással



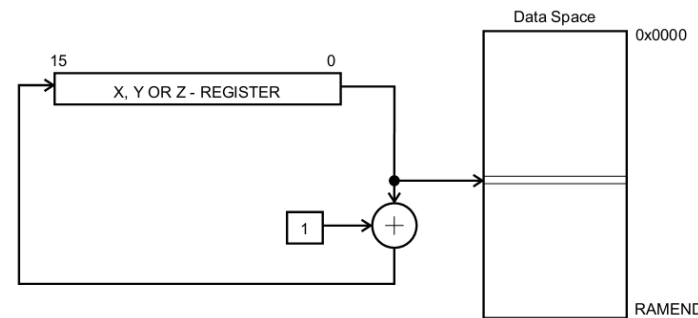
# Címzési módok V.

- Adat közvetett címzése előzetes csökkentéssel
  - Az adatbájt 16-bites címe az X, Y, vagy Z regiszterben helyezkedik el, és értéke eggyel csökken a művelet előtt. A forrás/cél regiszter Rr/Rd lehet. (Pl.: LD R16, -X)
- Adat közvetett címzése utólagos növeléssel
  - Az adatbájt 16-bites címe az X, Y, vagy Z regiszterben helyezkedik el, és értéke eggyel nő a művelet után. A forrás/cél regiszter Rr/Rd lehet. (Pl.: LD R16, +X)

Adat közvetett címzése előzetes csökkentéssel



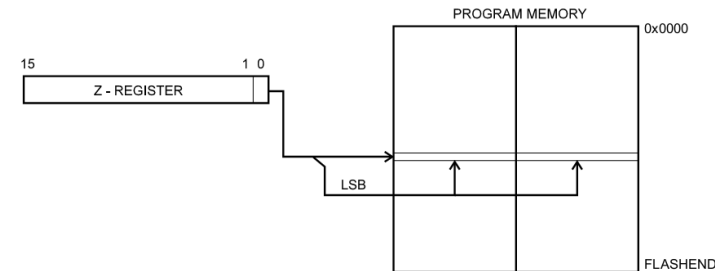
Adat közvetett címzése utólagos növeléssel



# Címzési módok VI.

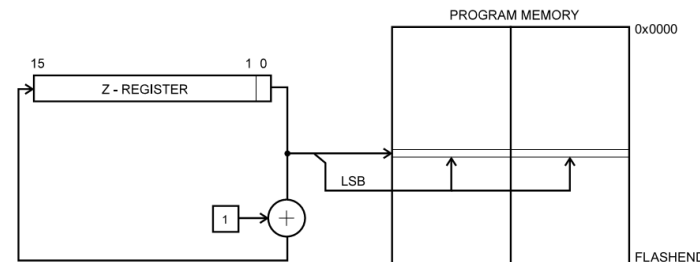
- Programmemória konstans címzése
  - A cím a Z regiszterben kerül megadásra, amelynek legkisebb helyiértékű bitje definiálja, hogy a felső vagy az alsó bájt kerüljön a Rd célregiszterbe. (Pl.: LPM R16,Z)

Programmemória konstans címzése



- Programmemória konstans címzése utólagos növeléssel
  - Ugyanaz mint az előző címzési mód, majd a címzés után a Z értéke eggyel növekszik. (Pl.: LPM R16, Z+)

Programmemória konstans címzése utólagos növeléssel

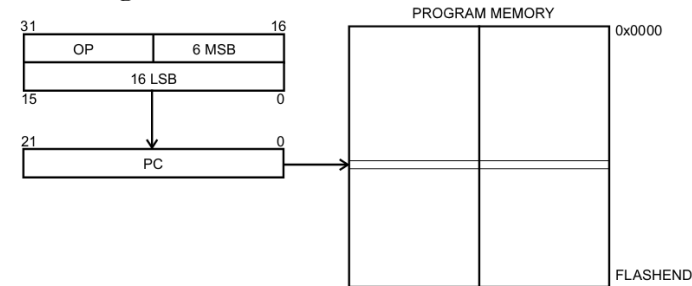




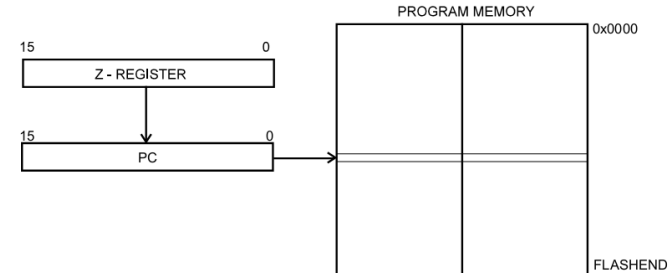
# Címzési módok VII.

- Programmemória közvetlen címzése
  - A program futása az utasítás az alsó 16 (22) bitjén megadott címen folytatódik. (Pl.: JMP ide)
- Programmemória indirekt címzése
  - A program futása a Z regiszterben megadott címen folytatódik. (Pl.: IJMP)
- Programmemória relatív címzése
  - A program futása a  $PC+k+1$  címen folytatódik. A  $k$  értékét az utasításban kell megadni -2048 és 2047 között (Pl.: RJMP 0x010)

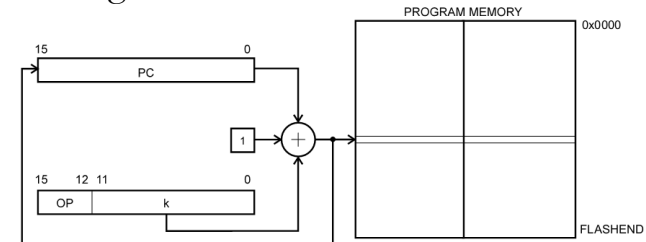
Programmemória közvetlen címzése



Programmemória indirekt címzése



Programmemória relatív címzése



# Utasítások fajtái

- Aritmetikai és logikai utasítások
  - Összeadás, kivonás, szorzás, törtszámok szorzása
  - ÉS, VAGY, Kizáró VAGY, komplement kód
  - Regiszter növelése, csökkentése, törlése, beírása
- Elágazások
  - Feltétel nélküli ugrások
  - Szubrutin és interrupt hívások, visszatérések
  - Feltételes ugrások az aritmetikai műveletek eredményének (SREG) megfelelően
- Adatmozgató utasítások
  - Adatmemória és regiszterek között
  - Programmemória és regiszterek között
  - I/O-k és regiszterek között
- Bit szintű utasítások
  - Eltolás, forgatás, alsó és felső 4 bit felcserélése
  - SREG, I/O, Rr regiszterek bitjeinek törlése, beírása
- MCU vezérlés
  - No-operation, sleep mode, watchdog reset
  - Break (csak a debugger számára)

# Vége

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

Köszönöm a figyelmet!