

Digitális rendszerek

Utasításarchitektúra szintje



Utasításarchitektúra

Jellemzők

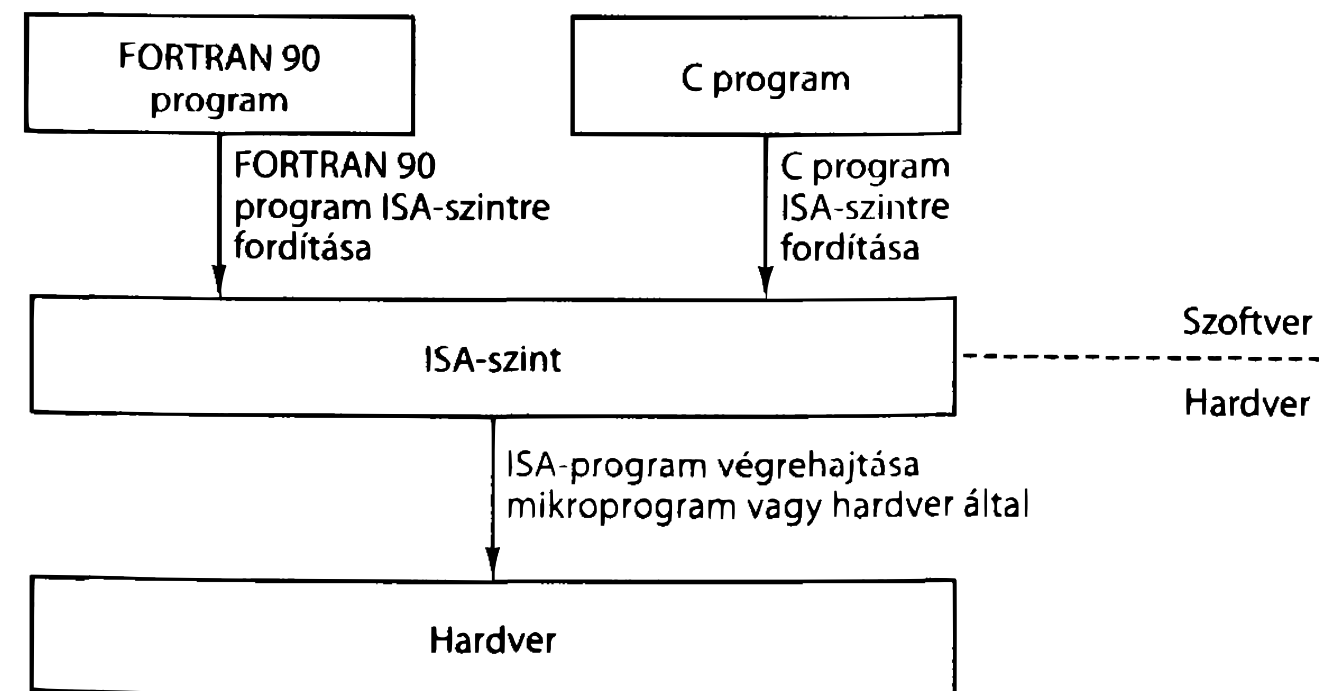
Mikroarchitektúra és az operációs rendszer közötti réteg

Eredetileg ez jelent meg először

Sokszor az assembly nyelvvel keverik

Megteremti a kapcsolatot a szoftver és a hardver között → a fordítóprogramok és a hardver közötti kapcsolatot definiálja

Visszafelé kompatibilitás fontos szempont lehet a tervezése során



Memóriaszervezési kérdések

Cellaméret

pl. 8 bit, szavak pl. 32 vagy 64 bit

Szavak igazítása

Visszafelé kompatibilitás költségnövekedéssel járhat

Egyesített vagy osztott címtartomány

Az adatok és az utasítások közös vagy különálló címtartományt használnak

Memóriaszemantika

Egy STORE A utasítást követő LOAD A mit fog visszaadni?

Memóriaműveletek végrehajtása: kötött vagy definiálatlan sorrendben

→ SYNC utasítás kiadása, vagy hardveres függőség kezelése

Regiszterek

Speciális célú regiszterek

Utasításszámláló, veremmutató, egyéb speciális funkciójú (gyorsítótárak, memória, I/O eszközök, stb. vezérlésére) regiszterek

Flags regiszter vagy Programállapotszó (PSW, Program Status Word) regiszter

Feltételbitek: ALU-ciklusok használják, utasítás eredményétől függ az állapotuk , pl.: N,Z,O,C,A,P

Egyéb jelzőbitek: nyomkövetés, CPU prioritási szint, INTA, stb.

Általános célú regiszterek

Lokális változók, részeredmények tárolása

Teljesen szimmetrikusak és felcserélhető is lehetnek

Ezekben belül egyesek speciális funkcióval is rendelkezhetnek

Utásításarchitektúra - példa

Pentium 4

IA-32 architektúra; MMX, SSE és SSE2 utasítások

Működési üzemmódok:

Valós, virtuális és védett

Szinte kimeríthetetlen címtartomány

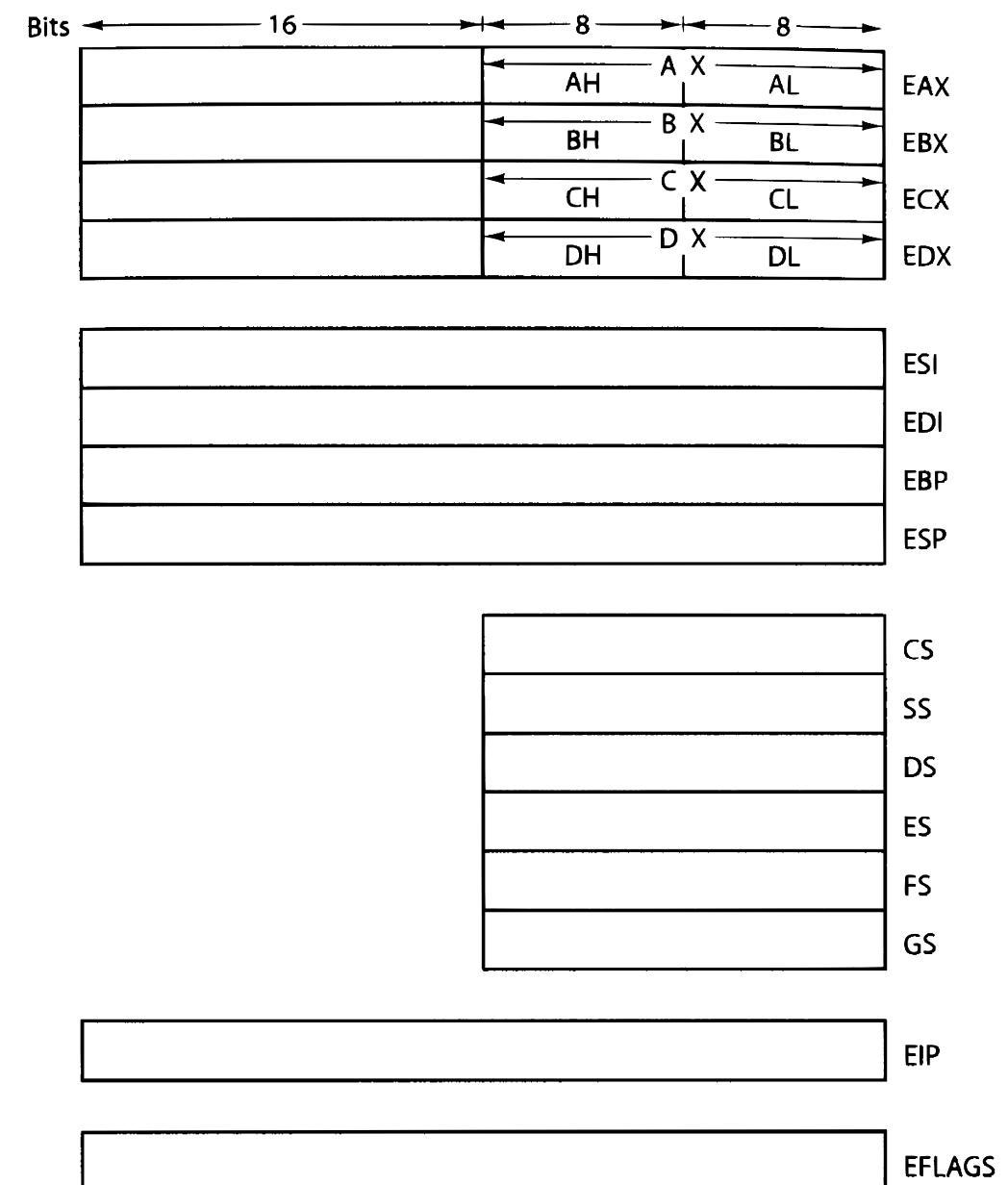
16384 szegmens, szegmensenként 2^{32} cím, a legtöbb op. rendszer (UNIX, Windows) csak egyet lát belőle

32 bites szószervezés

Tárolás: Little-endian szerint

32 bites regiszterek

EAX, EBX, ECX, EDX 8, 16, illetve 32 bites regiszterként is használható (visszafelé kompatibilitás)



Adattípusok

Numerikus

Egész számok: tipikusan 8, 16, 32 és 64 bites, előjeles vagy előjel nélküli

Lebegőpontos számok: leggyakrabban 32, 64 vagy 128 bites

Binárisan kódolt decimális (BCD) számok

Nem numerikus

Karakteres: ASCII (7bit), UNICODE (16bit)

Logikai: Boolean (1bit); gyakorlatban 1 bájt vagy 1 szó hosszúságú helyet igényel

Bittérkép: logikai értékek teljes tömbje, pl 32 bit = 32 logikai érték

Mutató: memóriacím tárolására

Adattípusok - példa

Pentium 4

Támogatja az előjeles és előjel nélküli egész, BCD és lebegőpontos számokat
8,16 bites egészek, 8 bites ASCII karakterek

Nem követeli meg az operandusok igazítását a memóriában (csak célszerű)

Típus	1 bit	8 bit	16 bit	32 bit	64 bit	128 bit
Bit						
Előjeles egész		X	X	X		
Előjel nélküli egész		X	X	X		
Binárisan kódolt decimális		X				
Lebegőpontos				X	X	

Utásításformátumok

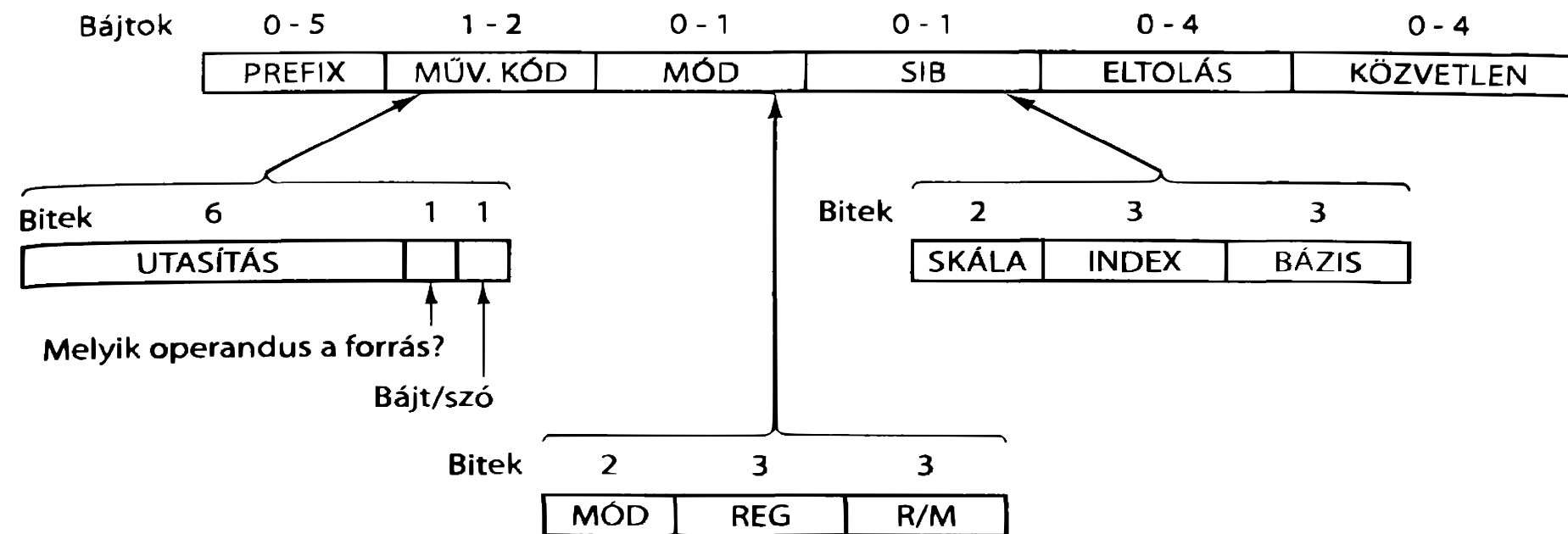
Utásítások felépítése

Általában minden utasítás követi a műveleti kód + operandus(ok) + eredmény helye struktúrárt

Címzés: az az általános módszer mely meghatározza az operandusok helyét (címét)

Leggyakoribbak a nulla, egy-, két- és háromcímes utasítások

Azonos vagy változó hosszúságú utasítások



Címzési módszerek

Közvetlen címzés

Az operandust nem címmel, hanem közvetlenül adjuk meg az utasításban

```
MOV R1, 4
```

Direkt címzés

Az operandust teljes címével adjuk meg

```
MOV R1, 0xe234h
```

Regisztercímzés

Az operandust nem címével, hanem egy regiszterrel adjuk meg

```
MOV R1, R2
```

Címzési módszerek

Regiszter-indirekt címzés

Memóriabeli operandus címe egy regiszterben található (mutató)

Előnye: memóriacímet nem az utasításban kell tárolni és végrehajtásonként változhat a cím

```
ADD R1, (R2)
```

Indexelt címzés

A memóriacímet egy regiszter és egy konstans értéke együttesen határozza meg.

```
MOV R4, A(R2)
```

Címzési módszerek

Bázis-index címzés

A memóriacím két regiszter és esetleg egy eltolási érték összegéből határozható meg

MOV R4, (R2+R5)

Veremcímzés

Egy lehetséges alkalmazás a nulla címes utasításformára (pl összes regiszter elmentése egy utasítással a verembe)

PUSHF → flagek verembe; POPF → flagek veremből

Utasítástípusok – Pentium4

Adatmozgató

Regiszter-memória, memória-regiszter és regiszter-regiszter irányú adatmásolás

Hétköznapi értelmezéstől eltérően a forráshelyen lévő adat változatlanul megmarad, annak egy másolata jön létre a célhelyen!

MOV, PUSH, POP, stb.

Aritmetikai és logikai

Alapműveletek, inkrementáló és dekrementáló, bitenkénti logikai műveletek

ADD, SUB, MUL, DIV, INC, DEC, AND, OR, NOT, XOR

Binárisan kódolt decimális

BCD számokon végzett műveletek utáni korrekciók

DAA, DAS

Utasítástípusok – Pentium4

Léptető és forgató

Bitenkénti léptetés vagy forgatás jobbra vagy balra

SAL, SAR, ROL, ROR

Tesztelő és összehasonlító

Két operandus eredménye a az EFLAGS regiszterben tárolódik

TST, CMP

Vezérlésátadó

Feltételes (a feltételbitek valamelyike alapján) és feltétel nélküli elágazás

Jxx, JMP, CALL, RET, IRET, LOOPxx

Utasítástípusok – Pentium4

Karakterlánc

Karakterláncokkal kapcsolatos műveletek

LODS, STOS, MOVS

Feltételkódok

EFLAGS egyes bitjeinek módosítása, a regiszter és verem közötti műveletek

STC, CLC, PUFHD, POPFD

Egyéb utasítások

Egyik előző kategóriába sem sorolható utasítás

SWAP, NOP, HLT, IN, OUT, WAIT