

Technologie procesoru

Intel 8086

OTÁZKY.....	1
LOGICKÁ STRUKTURA, PRINCIP ČINNOSTI, ZÁKLADNÍ CHARAKTERISTIKA, INSTRUKČNÍ SOUBOR.....	1
STRUKTURA REGISTRŮ, ADRESACE PAMĚTI, VSTUP/VÝSTUP.....	1
PŘERUŠOVACÍ SYSTÉM.....	1
TECHNOLOGIE PROCESORU – INTEL 8086.....	2
1.LOGICKÁ STRUKTURA, PRINCIP ČINNOSTI, ZÁKLADNÍ CHARAKTERISTIKA, INSTRUKČNÍ SOUBOR.....	2
<i>Základní vlastnosti.....</i>	2
<i>BIU.....</i>	2
<i>EU.....</i>	2
<i>Instrukční soubor.....</i>	3
<i>Instrukce.....</i>	3
<i>Kódování instrukcí.....</i>	4
STRUKTURA REGISTRŮ, ADRESACE PAMĚTI, VSTUP/VÝSTUP.....	4
<i>Struktura registrů.....</i>	4
<i>Všeobecné registry.....</i>	4
<i>Indexové a ukazatelové registry.....</i>	5
<i>Segmentové registry.....</i>	5
<i>Adresace paměti.....</i>	5
<i>Adresace operandů.....</i>	6
<i>Adresace operandů v paměti.....</i>	6
<i>Vstup/Výstup.....</i>	7
PŘERUŠOVACÍ SYSTÉM.....	7
<i>Přerušovací systém.....</i>	7
<i>Hardwarové provedení.....</i>	7
<i>Popis funkce.....</i>	7

Otázky

Logická struktura, princip činnosti, základní charakteristika, instrukční soubor

Charakteristika, Napájecí napětí
 Charakteristika sběrnice
 BIU, EU
 Struktura registrů, Registr příznaků
 Instrukční sada, skupiny instrukcí

Struktura registrů, adresace paměti, vstup/výstup

Doplňující otázky
 Logická adresa, Adresa absolutní, relativní
 Adresace operandů
 Vstup/Výstup

Přerušovací systém

Účel, HW realizace
 Přerušení (maskovatelné, nemaskovatelné)
 Tabulka vektorů přerušení, Průběh přerušení

Technologie procesoru – Intel 8086

1. Logická struktura, princip činnosti, základní charakteristika, instrukční soubor

Základní vlastnosti

Úplná 16ti bitová paralelní universální procesorová jednotka

Vyrobena unipolárními technologiemi HMOS

Pouzdro DIL 40

Ekvivalent cca **29 000 tranzistorů**

Napájení +5V, 275 mA max

Vstupy a výstupy TTL úrovně

Hodinový signál jednofázový

20ti bitová adresová sběrnice dovoluje adresovat 1M paměti

16ti bitová interní a 16 bitová (8mi bitová u 8088) externí datová sběrnice

Základem architektury jsou 2 zřetěžené subprocesory. Pracují poměrně nezávisle a využívají překrytí fáze zápisu, čtení i výběru instrukce s fázemi vykonání instrukce předchozí. Výsledkem je zvětšení rychlosti a zmenšení nároků na rychlost hlavní paměti.

BIU

Zajišťuje styk s vnější sběrnicí, výpočet adresy a asynchronní výběr instrukcí (realizuje fázi vykonej).

Funkčními bloky jednotky BIU jsou:

- 20 bitová sčítačka pro výpočet adresy (segment + offset = adresa)
- programový čítač, vnitřní komunikační registry
- 4 segmentové registry (CS, DS, ES, SS)
- logika řízení vnější sběrnice
- fronta připravených instrukcí (realizovaná jako 6ti bajtová paměť FIFO).

EU

Vykonává připravené instrukce.

Funkčními bloky jednotky jsou:

- ALU se třemi registry pro dočasné uložení operandů
- registr příznaků
- skupina registrů pro všeobecné použití (4 registry HL, dva registry ukazatelů, dva indexní registry)
- logika realizace instrukcí provedená jako mikroprogram
- logika řízení vnitřní sběrnice a vnitřní synchronizace, která mění vnější hodinový signál v posloupnosti interních řídicích impulsů.

Registr příznaků:

- má 9 jednobitových indikátorů uspořádaných v 16 bitovém slově:
- **CF** – přenos (carry)
- **PF** – parita výsledku je sudá (parity)
- **AF** – přenos z nižšího nibble AL registru (auxiliary)
- **ZF** – výsledek je nulový (zero)
- **SF** – nastaví se znaménko, když je záporné (sign)
- **OF** – aritmetické přeplnění (overflow)
- **DF** – autodekrementace (direction)
- **IF** – povolení přerušení (interrupt)
- **TF** – krokovací režim (trap)

Instrukční soubor

Instrukční soubor obsahuje 133 instrukcí, které dovolují zpracovávat jak 8-bitové tak i 16-bitové datové položky.

Instrukce může mít jeden, dva nebo žádný operand. Délka instrukce je proměnná v rozmezí 1 - 6 bajtů.

Prvý bajt vždy obsahuje instrukční kód, ostatní bajty nesou informaci o adresách operandů.

Prvý bajt také obsahuje speciální pole (bit W = 1 slovo) určující, zda instrukce zpracovává bait nebo slovo.

Procesor je schopen realizovat většinu známých adresových módů.

Instrukce

Skupina 0.

Základní podskupinu představují dyadické instrukce, aritmeticko-logické: ADD, OR, ADC, SVV, AND, SUB, XOR, CMP. Provedení je modifikováno: BYTE/WORD, Z/DO paměti a akumulátor s bezprostředním operandem. Celkem 48 instrukcí (6x8).

Druhou podskupinu tvoří 4 pomocné aritmeticko-logické instrukce: AAA, AAS, DAA, DAS (úpravy při práci s čísly ve formátu BCD)

Třetí podskupinu obsahuje 8 jednobitových instrukcí po uložení/vytažení obsahu segmentových registrů do/ze zásobníku (2x4).

Do poslední podskupiny patří jednobajtové prefixy pro změnu implicitního přiřazení segmentových registrů (4).

Celkem skupina 0 obsahuje 64 instrukcí.

Skupina 1.

Skládá se ze dvou podskupin.

Prvá podskupina je tvořena jednobajtovými instrukcemi nad šestnáctibitovými registry: INC, DEX, PUSH, POP (4x8=32 instrukcí).

Druhá podskupina obsahuje instrukce pro podmíněné skoky. Instrukcí je 16 (2x8) a jsou dvoubajtové.

Druhý bajt tvoří konstanta (+- 128). Jedná se tedy o instrukce pro krátký skok v rámci segmentu. Adresa skoku je relativní k IP (IP+-128).

Skupina 2.

Velmi komplexní, skládá se z osmi podskupin.

Podskupinu 0 tvoří dyadické, aritmeticko-logické operace s bezprostředním operandem ADD, OR, ADC, SBB, AND, SUB, XOR, CMP. Provedení je opět modifikováno Byte/Word s tím, že u aritmetických operací může být bezprostřední operand typu BYTE, ale operace je WORD.

Podskupinu 0 (20x) tedy tvoří dyadické, aritmeticko-logické operace s bezprostředním operandem.

Co se má provést je kódováno polem REG druhého bajtu (instrukce pracují s bezprostředním operandem a toto pole je tedy nevyužito).

Provedení je opět modifikováno BYTE/WORD s tím, že u aritmetických operací může být bezprostřední operand typu BYTE, ale operace je WORD (u logických operací to není možné). Využity jsou jen některé kombinace (200, 201, 203) některé jsou nelogické (202) a zbyte jsou obsazeny instrukcemi TEST a XCHG.

Podskupina 1 (21x) realizuje přesun dat mezi registrem a pamětí. Opět modifikace BYTE/WORD. Jako registr může být použit i registr segmentový (214 a 216).

Podskupina 2 (22x) obsahuje instrukce typu WORD, výměny položek mezi akumulátorem a jiným registrem (XCHG). Kombinace 220 má zvláštní význam a interpretuje se jako NOP (XCHG AX, AX nemá smysl). Přesuny pokračují v podskupině 4.

Podskupina 4 (24x). Akumulátorové provedení (240 – 243) s přímou adresací operandu v paměti.

Zbývající kombinace kódují instrukce přesunu a porovnání řetězců znaků (244 – 247).

Podskupina 5 (tedy 25) definuje operace s řetězcí znaků: STOS, LODS, SCAS. Modifikace BYTE/WORD.

Podskupina 6 (24) obsahuje instrukce přesunu bajtových datových položek (bezprostřední operand) do registrů (8 bitové registry pro všeobecné použití)

Podskupina 7 (27x) totéž s tím rozdílem, že se jedná o slova (a 16ti bitové registry)

Skupina 3.

Podskupina 2 (tedy 32x) obsahuje operace posunů. V prvním bajtu je zakódován způsob provedení posunu: BYTE/WORD, jednou/vícekrát a pole REG druhého bajtu specifikují operaci (doleva, doprava, logicky, aritmeticky, rotace). Princip stejný jako ve skupině 20x.

Podskupina 4 (tedy 34x) kóduje instrukce pro iterace, modifikace instrukce LOOP, 340-343 a operace vstupu/výstupu (IN, OUT, 344-347). Podskupina 5 (35x), definuje instrukce pro předávání řízení s návratem (CALL) a bez návratu (JMP). Podskupina 6 (36x) zahrnuje jednak prefixy opakování instrukce (REP), jednak komplexní aritmetické operace (násobení a dělení s/bez znaku).

Podskupina 7 (37x) jsou jednobajtové instrukce pro manipulace s příznaky (nastavení, nulování). Zcela zvláštní postavení má sekvence 33x (nazývaná ESC). V zásadě říká, že kód, který následuje není určen přímo procesoru, ale má jej vykonat jiná (externí) jednotka (koprocessor).

Kódování instrukcí

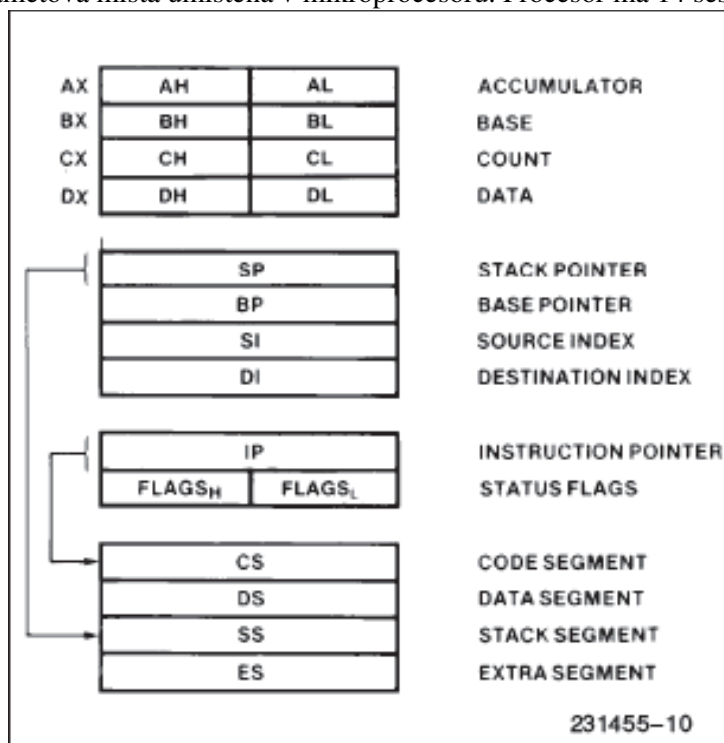
u tohoto procesoru neexistuje vztah mezi mnemonikou instrukce a její číselnou reprezentací. Důvody jsou 2:

- procesor má komplexní instrukční sadu (CISC). Vykonání instrukcí je složitá posloupnost úkonů a je proto řízená mikroprogramem, nikoli dekodovací logikou.
- instrukční sada byla navržena tak, aby výsledný kód byl co nejkratší. Často používané instrukce jsou kódovány takovou formou, aby byly krátké. Tedy cosi jako komprimace. Instrukce mají proměnnou délku (16 bajtů).

V zásadě lze vydělit dvě základní skupiny instrukcí: Jednobajtové a vícebajtové. OP kód je vždy první bajt nebo první dva bajty.

Struktura registrů, adresace paměti, vstup/výstup**Struktura registrů**

Registry jsou paměťová místa umístěná v mikroprocesoru. Procesor má 14 šestnácti bitových registrů

**Všeobecné registry**

AH, AL	střadač (akumulátor)
BH, BL	báze
CH, CL	čítač
DH, DL	data

Lze je používat jako dvojice 8mi bitových registrů AH je pro vyšších 8 bitů, AL je pro nižších 8 bitů Tyto registry se používají převážně pro aritmetické operace.

Speciálně je pro ně určen registr **AX** (akumulátor), řada instrukcí pracuje převážně jen s tímto registrem.
BX se používá především k výpočtu adresy v některých adresových módech.
CX je čítač, používá se u instrukcí s opakováním.
DX je datový registr a nemá speciální funkci.

Indexové a ukazatelové registry

SP ukazatel zásobníku
BP ukazatel báze
SI index zdrojové adresy
DI index cílové adresy

Tyto registry se nejčastěji používají pro adresaci dat.

Indexové registry mají specifikovaný způsob využití.

SP (ukazatel zásobníku) obsahuje hodnotu offsetu zásobníku mikroprocesoru a používá se při práci se zásobníkem a voláním podprogramu.

BP je určen k adresování dat hlavní paměti. Při spolupráci s vyšším programovacím jazykem je využíván jako ukazatel na parametry volané procedury.

SI a **DI** se využívají při přenosech boku dat a při přístupu do paměti.

Segmentové registry

CS segmentový registr programu (Code Segment)
DS segmentový registr dat (Data Segment)
SS segmentový registr zásobníku (Stack Segment)
ES pomocný registr (Extra Segment)
IP instruction pointer

CS je nejdůležitější, jelikož obsahuje segmentovou část adresy právě běžícího programu.

DS je určen k adresování dat v hlavní paměti, tzn., že pokud instrukce pracuje s daty v hlavní paměti, je určen právě tento registr.

SS ukazuje na segment strojového zásobníku.

SS a **SP** je přesná adresa strojového zásobníku v hlavní paměti. Jde využít u instrukcí, které pracují se zásobníkem.

ES je využíván při přesunech dat, jinak je volně k dispozici.

IP - tento registr obsahuje offsetovou část adresy právě zpracovávané instrukce.

Adresace paměti

Mikroprocesor pracuje s 20 bitovou adresou.

Nejmenší adresovatelnou jednotkou je bait.

Primární (operační) paměť má tedy kapacitu 1 048 576 baitů.

Architektura předpokládá, že paměť je přístupná v logických blocích – segmentech.

Segment je logická konstrukce – fyzicky je paměť souvislý lineární prostor.

Segment je blok paměti o délce 64 kB.

Počátek segmentu je definován obsahem příslušného segmentového registru.

Adresa je 20ti bitová, segmentový registr je sice fyzicky 16ti bitový ale logicky je až 20ti bitový – zcela vpravo jsou 4 nevyjádřené bity (nejmenší krok v segmentovém registru je 16).

Adresa segmentu vždy leží na adrese na adrese dělitelné 16ti.

Procesor má 4 segmentové registry.

V daný okamžik má přístup k 4 x 64.

Segment je logická konstrukce – segmenty se mohou navzájem částečně překrývat.

Adresu položky v primární paměti lze vyjádřit několika způsoby:

- Logická adresa je dvousložková a zapisuje se jako dvě 16ti bitové hodnoty **segment : offset**
- Lineární adresa se z adresy logické získá posunutím segmentové části o 4 bity doleva (násobením 16ti) a přičtením offsetové části adresy

- Finální je adresa fyzická – hodnota, která je vystavena v registru MAR. Na procesoru 8086 se lineární adresa rovná fyzické. Přepočít logické adresy na lineární provádí BIU.

Členění operační paměti na segmenty není samoučelné, protože vychází z programátorského pohledu na paměť.

Programátor předpokládá, že program zavedený do paměti bude rozdělen na text(kód), data(proměnné a konstanty) a zásobník.

Jednosložková adresa(lineární) je adresa absolutní.

Dvousložková adresa je relativní způsob adresace – offset je relativní vzdálenost od počátku segmentu.

V momentě kompilace programu nikdy nevíme na jakou adresu v paměti bude při vykonání program zaveden.

Adresy proměnných – relativní od začátku bloku na kterém jsou uloženy.

Po spuštění programu bude blok proměnných vložen do datového segmentu a správná absolutní adresa všech proměnných se získá nastavením jediné hodnoty – adresou v segmentovém registru. Dvousložková adresa řeší i problém bezpečnosti.

Adresace operandů

mikroprocesor umožňuje adresovat operandy v instrukcích 24 způsoby.

Operandy mohou být v registrech, paměti, I/O registrech(portech) nebo být součástí instrukce. Pro dynamické instrukce platí: pouze jeden operand může být uložen v paměti a výsledek první z operandů přepíše.

Adresace operandů v paměti

Adresa paměťového místa je dvousložková ve tvaru segment : offset.

Instrukce pracují s pamětí, tedy používají relativní (logickou) adresu.

Paměťový subsystém naopak používá adresy absolutní.

Přepočít adresy relativní na absolutní provádí BIU.

Segmentové registry se přiřazují implicitně a v instrukcích se proto neuvádějí.

Existuje však možnost, dané přiřazení přepsat předřazením jednobajtového prefixu změny segmentového (tvar 00188110).

Program vždy pracuje pouze s 16ti bitovými adresami.

Offsetová část adresy operandu je obecně vícesložková.

Může se skládat z posunutí, báze a indexu.

Posunutí je konstanta, která je součástí instrukce. Má velikost jeden nebo dva bajty.

Hodnota indexuje v určeném indexovém registru a hodnota báze v registru báze.

Obsah registru je proměnný; konečnou platnost má hodnota která je v těchto registrech uložená v momentě vykonání instrukce.

EU procesoru v příslušné fázi realizaci instrukce sečte jmenované hodnoty(posunutí + báze + index) a získá finální hodnotu offsetu – efektivní adresu, kterou předá BIU.

Ta posune a přičte obsah příslušného segmentového registru a získá adresu lineární.

Možné kombinace:

- pouze posunutí(přímá adresace)
- pouze index
- pouze báze
- báze a posunutí
- index a posunutí
- báze a index a báze.

Zřejmým důsledkem kompilovaného způsobu získání offsetové části adresy je prodloužení doby vykonání instrukce o cykly potřebné k výpočtu efektivní adresy.

Jedinou výjimkou z pravidla o jednom operandu v paměti je instrukce pro přesun řetězce znaků(bloků dat).

Odkud se data přesouvají(zdroj) udává dvojice DS:SI, místo přesunu(cíle) je určeno ES:DI. Instrukce sama nemá operandy – jsou dány obsahy registrů.

Vstup/Výstup

Mikroprocesor může pro styk s vnějšími zařízeními používat V/V brány adresování odděleně od hlavní paměti (isolated IO). Může komunikovat buď po 8 nebo 16 bitech. Má přístup k 64K 8mi bitových nebo 32K 16ti bitových bran. Pro adresování bran (sudá-lichá) platí stejná doporučení a omezení jako při adresování paměti.

Přerušovací systém

Přerušovací systém

Přerušování je obecně asynchronní událost, která způsobí pozastavení probíhajícího procesu a umožní spuštění procesu jiného.

Zdroj signálu inicializujícího přerušování může být:

- Vnější (třeba z periferie)
- Vnitřní (softwarové přerušování)

Po příchodu signálu přerušování je běžná sekvence instrukcí dočasně přerušena řízení je předáno rutině obsluhy (interrupt handler)

Obsluha přerušování vyžaduje provedení určitých operací navíc (režie přerušování) a to zpotřebovává jak paměť tak i čas.

Nákladné je hlavně zabezpečení neměnnosti pracovního prostředí přerušovaného procesu, tak aby byla zajištěna kontinuita procesu.

Určitou dobu také trvá samostatná operace přepnutí procesů.

Každý zdroj přerušování má svou obslužnou rutinu.

Adresy vstupních bodů těchto rutin se nazývají vektory přerušování a jsou uspořádány do tabulky uložené v paměti od adresy nula.

Vnější přerušování z různých zdrojů mohou běžně vzniknout ve stejný časový okamžik.

Pořadí důležitosti v jakém se přerušování obsluhuje určuje priorita přerušování a je řízena hardwarově.

Hardwarové provedení

Procesor 8086 má pro příjem signálů vnějšího přerušování 2 vývody.

Vývod 17, označovaný jako MNI(nemaskovatelné přerušování) slouží k signalizaci fatálních událostí.

Signál nelze ignorovat(zamaskovat, odstínit), protože oznamuje katastrofické stavy systému(chyba parity paměti, sběrnice)

Procesor na ně musí okamžitě reagovat.

Na vývod 18, označený jako INTR, je připojen programový kontrolér přerušování(PIC).

Má 8 stupňů a obvody lze řadit do kaskády.

Další přerušování realizované hardwarově slouží k ladění.

Nastavením TF v registru příznaků přijde procesor do krokovacího režimu, tj. při každém vykonání instrukce dojde k přerušování.

Obslužná rutina aktivuje ladící systém, který stanoví co se vykonáváním instrukce změnilo.

Další čistě HW přerušování je reakce na dělení nulou a přetečení.

Tabulka rezervovaných vektorů přerušování obsahuje 32 vektorů.

Popis funkce

Při vzniku přerušování se provedou následující operace:

- Dokončí se rozpracovaná instrukce, přerušování se uplatní až po dokončení instrukce.
- Uloží se do zásobníku registr příznaků.
- V registru příznaků se vynulují bity IF a TF. Do zásobníku se uloží registr CS a registr se naplní z adres $N*4+2$. Do zásobníku se uloží IP a registr se naplní $N*4$

Programátor v rutině obsluhy přerušování musí uschovat obsah všech registrů, které hodlá použít. Přerušovací systém je možno při obsluze přerušování aktivovat a dovolit přerušování obsluhy přerušování.

Návrat k přerušovanému procesu se provede instrukcí IRET.

Ta obnoví ze zásobníku IP, CS a registry příznaků.