

**** MT - MO - OSNOVA ****

Zpracoval Michal Frdlík

Obsahuje komplet všechno, jestli si rozumíme

Je vhodná pro běžné humanoidy od tří let

1a) Základní pojmy počítačového systému, současné víceúrovňové p. systémy

- Počítač je sekvenční automat, jehož chování je přesně dáno programem, uloženým v jeho paměti.
- Činnost počítače je řízena instrukcemi, jejichž posloupnost se nazývá program
- VON NEUMANNHOVÉ schéma p.: IN-D>MEM-D>OUT; CPU-R>IN, MEM, OUT; CPU-D>MEM
- Pravidla zápisu instrukcí tvoří jazyk - strojový jazyk, v němž jsou instrukce zadávány
- Stavový vektor: Množina veličin, tvořících stav procesu
- Je-li jazyk J2 na vyšší úrovni než J1, je nutno J2 kód do J1 kódu přeložit a to se děje pomocí:
 - KOMPILACE: Celý J2 program se pomocí překladače - kompilátoru - přeloží do J1 a uloží se do paměti, odkud se následně spouští
 - INTERPRETACE: Program zvaný interpret převádí jednotlivé J2 instrukce na J1 instrukce a ihned je provádí
- Vyšší jazyky běží na virtuálních počítačích a z pohledu uživatele obsluhujícího virtuální počítač je jazyk tohoto počítače jazykem strojovým
- Úroveň 3 je o-počítač a úroveň 4 a-počítač, který používá jako strojový jazyk Assembler

1b) Blokové schéma a popis 8086

AX - akumulátor, BX - báze, CX - čítač, DX - data
SP - stack pointer, BP - base pointer, SI - source index, DI - destination index - dvakrát po osmi na vnitřní sběrnici a z ní dvakrát po osmi na operandy (interní registry procesoru), z nich dvakrát po osmi na ALU, která po devíti na indikátory, po šestnácti zpátky na vnitřní sběrnici a po osmi doprava na forntu, která je osmibitová a má 6 polí; z fronty na 20 bitovou sčítačku, ta nahoru po šestnácti na CS - code segment, DS - data segment, SS - stack segment, ES - extra segment a IP - instruction pointer; ze scítačky 20/16 na vnější multiplexní sběrnici

2a) Úroveň konvenčních sálových p. a p. PC

- Vznikají téměř vždy díky vrstvě, postavené na hardwaru a jsou v základě dva typy - kanálové a sběrnicevé
- Sběrnicevé počítače obsahují univerzální sběrnici, na kterou je připojena CPU, Paměť, Vstupy a Výstupů
- Jsou připojeny pomocí tří sběrnic - Řídící, Adresové a Datové
- Datovou sběrnici protékají data mezi komponentami počítače
- Pomocí Adresové sběrnice procesor adresuje komponenty, se kterými komunikuje
- Pomocí Řídící sběrnice procesor ovládá své okolí a přijímá od něj požadavky a zpětnou vazbu
- Tento typ počítačů není schopen zařídit na úrovni k multitask, což lze prokázat dvojím způsobem: a) V jeho zapojení nacházíme pouze jedno von Neumannovo schéma počítače

- b) Více napětí v jednom čase není na sběrnici možná situace
- Nejtypičtějším příkladem těchto p. jsou p. PC
 - VON NEUMANNOVO schéma p.: IN-D>MEM-D>OUT; CPU-R>IN, MEM, OUT; CPU-D>MEM
 - *KANÁLOVÉ POČÍTAČE*: Sestávají z jedné velké hlavní paměti, společné pro všechny kanály, z n kanálů, které mají vlastní malou paměť, vlastní řadič (procesní jednotku) a vlastní IO brány.
 - Každý kanál tvoří princip jednoho von Neumannova schématu a tudíž tento počítač dovede na úrovni k zařídit multitask, jehož počet paralelně probíhajících procesů závisí na počtu n kanálů
 - Jsou zde běžné programy malé, které se vejdu do vlastní paměti kanálu ... běžně je program rozdělen na více částí a zpracováván jinými kanály

2b) Signály procesoru 8086

- Výstupní:
 - A - Adresová sběrnice (vstupně výstupní)
 - D - Datová sběrnice (vstupně výstupní)
 - RD - CPU přijímá data ze sběrnice
 - LOCK - Vyhrazení řízení lokální sběrnice pro CPU
 - QSI - Stav fronty
 - QSB - Taktéž
 - M/IO - Rozlišuje takty s přístupem k paměti a k IO
 - WR - Data na sběrnici jsou platná výstupní
 - INTA - Přijetí žádosti o přerušeni
 - ALE - Přepisuje adresu do adresového registru
 - DT/R - Řízení orientace průchodu obousměrnými budiči
 - DEN - Řízení obousměrných budičů
 - HLDA - CPU potvrzuje přijetí požadavku HOLD a oznamuje přechod sběrnice do 3. stavu
 - BHE - Řízení přenosu slabik
- Vstupní:
 - HOLD - Požadavek nadřazeného účastníka pro uvolnění sběrnice
 - READY - Oznamuje CPU, že data na sběrnici jsou platná výstupní
 - INTR - Maskovatelný požadavek o přerušeni
 - TEST - Signál testovaný instrukcí WAIT - 0 - pokračuje na další instrukci, 1 - setrvává v Tw
 - NMI - Nemaskovatelný požadavek o přerušeni
 - RESET - Ukončuje aktivitu CPU a předává řízení instrukci na adrese 0xFFFF0
 - CLK - Hodinový signál, 1 fázový, 33% střída
 - MN/MX - Ve kterém módu CPU pracuje

3a) Úroveň konvenčních sběrnicevých počítačů

- Vznikají téměř vždy díky vrstvě, postavené na hardwaru a jsou v základě dva typy - kanálové a sběrnicev
- Sběrnicev počítače obsahují univerzální sběrnici, na kterou je připojena CPU, Paměť, Vstupy a Výstupů
- Jsou připojeny pomocí tří sběrnic - Řídící, Adresové a Datové
- Datovou sběrnici protékají data mezi komponentami počítače
- Pomocí Adresové sběrnice procesor adresuje komponenty, se kterými komunikuje
- Pomocí Řídící sběrnice procesor ovládá své okolí a přijímá od něj požadavky a zpětnou vazbu
- Tento typ počítačů není schopen zařídit na úrovni k multitask, což

lze prokázat dvojím způsobem: a) V jeho zapojení nacházíme pouze jedno von Neumannovo schéma počítače

b) Více napětí v jednom čase není na sběrnici možná situace

- Nejtypičtějším příkladem těchto p. jsou p. PC

- VON NEUMANNOVO schéma p.: IN-D>MEM-D>OUT; CPU-R>IN, MEM, OUT; CPU-D>MEM

3b) Přerušovací systém 8086

- *Externí přerušování*: Vstupní signál *nemaskovatelného* přerušování INTR se budí programovatelným řadičem přerušování 8259A. Maskovatelný je proto, že lze akceptování přerušování předem zakázat nastavením IF - interrupt flagu instrukcemi STI a CLI. Procesor potvrzení požadavku potvrzuje dvěma cykly signálu INTA. Řadič přerušování tento signál přijme a umístí na datovou sběrnici číslo, označující typ přerušování. Procesor toto vynásobí čtyřmi a dostane tak adresu v tabulce vektoru přerušování. V obsluze přerušování se pak vrací zpět instrukcí IRET, který opět povoluje IF a vrací zpět uloženou konfiguraci procesoru. Signál INTR není zachycován registrem a musí proto být udržován až do doby, kdy procesor potvrdí přijetí.

Maskovatelné přerušování NMI se do registru zachycuje. NMI nebere v úvahu IF a provede se absolutně vždy! Mnohdy totiž signalizuje katastrofické scénáře, jejichž ošetření nesnese prodlení.

- *Interní přerušování*: a) Instrukcí INT n (n = 0 - 255). Pomocí této instrukce lze uměle vyvolat kterýkoliv přerušovací podprogram.

b) Instrukcí INTO, která vykoná INT 4, pokud nastane overflow - čili pokud OF bude 1.

c) Po dělení nulou dojde k přerušování typu 0

d) Pokud je TF=1 (trap flag), dojde k přerušování typu 1 po každé vykonané instrukci. Je to krokový režim, používaný při debugingu.

- Vnitřní prioritě mají interní přerušování, potom následuje NMI, INTR a nejnižší prioritě má krokový režim

4a) Formáty instrukcí k-jazyka, jejich adresování, data

- *Formát instrukce* je názorné vyjádření, které má za úkol popsat, jak instrukce vypadá a slouží k vytvoření dekodéru instrukcí

- Sestává z OZ - operačního znaku a žádného, jednoho či více operandů; v bitovém vyjádření instrukce je OZ vždy jako první

- Adresování: Adresování přímého operandu - MOV AX, 4

Přímé adresování - MOV AX, [4]

Nepřímá adresace - MOV AX, [[4]]

Adresování s registrem - MOV AX, [BX+SI+4]

Adresování s indexem - MOV AX, [4+SI]

- *Zásobník* je paměť typu SAM, u níž poslední vložená hodnota, bude vybrána jako první - LIFO - Last In, First Out

- Opakem je fronta FIFO - First In, First Out

- k-data - čísla se znaménkem, bez znaménka - znaménkový bit, doplněk

4b) 8086 v minimálním módu

- Při minimálním módu signál MN/MX = Ucc

- Takt procesoru je rozdělen na intervaly T1, T2, T3, (TW), T4 a TW může být libovolné množství

- Signál M/IO signalizuje, zda jde o operace s pamětí či se vstupy nebo výstupy

- Signál Tn trvá jednu periodu signálu CLK

- V době T1 se aktivuje signál ALE - Address Latch Enable a do

vnějšího záchytného registru se převezme adresa a signál BHE

- V době T2 adresové vývody procesoru přecházejí do 3. Stavů
- Jde-li o cyklus čtení, aktivuje se v T2 signál RD, adresované zařízení pošle data na sběrnici a oznámí to signálem READY; procesor data převezme a vypne RD, takže adresované zařízení přepne své vývody do 3. Stavů
- Jde-li o cyklus zápisu vysílá data určená k zápisu na adresované místo, kde zůstávají až do poloviny T4 - Během T2 aktivuje signál WR

5a) CIKAP

- délka slova 16 bitů, HP = 2^{13} slov, adresa 13 bitů, data 16 bitů
- zásobníkový počítač se zásobníkem v HP, aritmetickými operacemi, k-jazyk
- instrukce PUSH, POP, SKO (skok na adr v RI), SZA (vyjmutí vrcholu zásobníku T a skok na adr je-li <0), SNU (je-li $=0$), SNZ (>0), SPR (vloží návratovou adresu do zásobníku a skok na adr), SNA (vyjmout návratovou adr ze zásobníku a skok na ni), PLU (POP T; POP S; PUSH S+T), MIN, NAS, DIV

5b) 8205

- DEKODODĚR - rychlý dekodér binárního kódu na kód 1 z 8 s negovanými výstupy; má vstupní přívody A0, A1, A2 na binární kód a osm výstupů 00 až 07
- Laicky řečeno to znamená: „Přivedeš binární kód na áčka a jedno z óček bude aktivní vole!“ - promiňte mi tu vsuvku, ale jsou tři minuty po půlnoci a já tady místo spaní mastím tyhle ptákoviny ;).

6a) HOMIP

- Na HOMIPu běží interpret CIKAPu
- HOMIP se skládá z 15 registrů, 39 datových cest, 16 bitové dvojkové sčítačky a posouvače o 1 řád vlevo
- Registry HOMIPu jsou a) s proměnným obsahem - b) s konst. Obsahem
- ad a) ČI - čítač instrukcí - 13 bit
 - SP - stack pointer - 13 bit
 - RI - registr instrukcí - 16 bit - OZ a ADR
 - AR - adresový registr - 13 bit
 - DR - datový registr - 16 bit
 - A, B, C, D - pracovní registry
 - X - univerzální čítač
- ad b) +1, -1, +0, 15, MINUS

6b) 8216

- OBOUSMĚRNÝ BUDIČ sběrnice, 4 bitový
- 4 dvojice opačně orientovaných 3 stavových zesilovačků (malý zesilovač = zesilovaček, k?)
- Lze jimi realizovat před mezi obousměrnou a dvěma jednosměrnými sběrnicemi

7a) Posloupnost provádění mikro operací HOMIPu

- mikrooperace se provádějí otevíráním a zavíráním datových cest mezi registry a funkční jednotkou, nebo mezi registry a registry
- funkční jednotky nemají paměťové prvky, ihned ukládat
- příklad PLU:
 - AR:=SP;
 - DR:=HP(AR);
 - A:=DR;
 - SP->EL; -1->EP;
 - SP:=SP+(-1);

```
AR:=SP;
DR:=HP(AR);
A->EL; DR->EP;
DR:=A+DR;
AR:=SP;
HP(AR):=DR;
```

7b) 8286

- *OBOUSMĚRNÝ BUDIČ sběrnic*, 8 bitový
- Trvale spojené vstupy a výstupy na obou stranách z každé dvojice sdružených zesilovačků, takže je méně univerzální

8a) Mikro jazyk HOMIPu

- má pouze 2 40 bitové instrukce OTEVRI_RB a TEST_A_PODM_SKOK
- OTEVRI_RB - nultý bit - 1 - OZ - zbylé pak reprezentují číslo řídicího bodu, který se má otevřít
- TEST_A_PODM_SKOK - instrukce se skládá z OZ 0, potom z několika bitů, které představují registr, který se má testovat, potom bity, které specifikují který řád se má testovat, potom na co se má testovat a potom adresu, na kterou se má skočit, posledních několik bitů je nevyužitých

8b) 8282

- REGISTR bez nulovacího vstupu, zato s výkonnými výstupními zesilovačky
- Používá se pro demultiplex sběrnic a pro buzení adresových sběrnic

9a) Pomocný jazyk HOMIPu a jeho interpreter

- psát programy pomocí posloupnosti 40 bitových čísel není ok
- mikrojazyk dokáže všechno, příkazy oddělené středníkem
- příkazy na jedné řádce se dějí během jednoho taktu
- přířazovací operátor „:=“, použitý sčítačky „->EL ->EP, INVERZE, L_POSUV“
- testování „if RI(bit)=1 then goto navesti“
- možno zmínit INTERPRET CIKAPU ... možná nutno

9b) Paměti počítačů

- Paměti počítačů mohou být implementovány číslicovými paměťovými obvody (jako hlavní paměti), pevnými disky či diskovými poli, vnějšími paměťovými zařízeními a úložnými paměťovými médii
- Účelem (svatým posláním) paměti je uchovávat data, což dělají prostřednictvím dvojkové soustavy, protože je jednoduchá
- Základní jednotka paměti je bit - čili dvojkový řád (0,1)
- Základní adresovatelná jednotka je byte - bajt - velikostí osmibitový, říct něco o binárních předponách
- Vnitřní paměti se dělí na: ROM - Read Only Memory

- Malé paměti s perzistentním

obsahem, jako je například BIOS nebo různé identifikační údaje různých zařízení

RAM - Random Access Memory

- Rychlá přepisovatelná paměť,

která uchovává informace po dobu jejího napájení a používá se jako operační paměť počítačů

10a) OS - základní pojmy

- *k-programy*: operační systém, supervizor, monitor, exekutiva
- počítač, který používá tyto programy je počítačem třetí úrovně, nebo o-počítač - operační počítač

- o-jazyk používá většinu instrukcí k-jazyka, kromě privilegovaných instrukcí, které jsou vyhrazeny pouze systému na k-úrovni - tedy operačnímu systému, supervizoru, monitoru, exekutivě
- operační systém pomáhá uživateli řešit:
 - Ovládání vnějších zařízení
 - Organizaci dat na vnějších médiích
 - Efektivní využívání počítače
 - Zadávání úkolů počítači
- operační systém patří k systémovému programovému vybavení počítače, který dodá společně s počítačem dodavatel
- různé OS, konkureční boj, nezáměr na sjednocení
- generace počítačů: 46-53 > Elektronky, sestavovací program, knih.
 - 54-62 > Tranzistory, monoprogramní OS
 - 63-70 > IO, multiprogramní OS
 - 71-78 > lepší IO, rozšiřitelné OS

10b) 8251A

- *OBVOD SÉRIOVÉ KOMUNIKACE*
- Při sériové komunikaci se jednotlivé bity slova přenášejí za sebou
- Náš obvod se také označuje zkratkou *USART* - Universal Serial Asynchronous Receiver Transmitter
- Lacičky řečeno: „Přijme paralelní, vysílá sériový; přijme sériový, vysílá paralelní“
- Obsahuje registr datové sběrnice, na které je připojena řídicí logika s *RESET*, *RD*, *WR*, *CLK* a *CS*, to jde na sběrnici, na kterou jsou připojeny registr vysílače a registr přijímače, které mají také vlastní řízení (oba dva!)

11a) Organizace využití počítače

- organizace využití závisí na druhu využívání
- o-program > *JOB*
- část o-programu > *TASK*
- posloupnost o-programů > *BATCH*
- některé počítače nechávají provádět několik částí o-programů najednou -> *MULTITASKING*
- OS - a) *Monoprogramní*
- b) *Multiprogramní*
- ad a) - na počítači běží jeden o-program po druhém
- ad b) - na počítači běží více o-programů nezávisle na sobě
- nepřímý přístup: uživatel zadává zakázku obsluze počítače a ta zadá práci počítači, který ji zpracuje; uživatel si hotovou práci vyzvedne na výdeji zakázek; doba obrátky je interval mezi příjmem a výdejem zakázky.
- *přímý přístup*: uživatel pracuje s počítačem přímo; pracuje s o-počítačem a přidělen určitý strojový čas; v takovém případě je možno jeden k-počítač přidělit víceru o-počítačů a tedy víceru uživatelů
- formou přímého přístupu je *přístup přes terminál*, který může být místní nebo vzdálený
- další formou přímého přístupu je *přístup interaktivní*, kde si uživatel s počítačem vyměňuje zprávy a na základě toho probíhá práce

11b) 8253 a 8254

- *PROGRAMOVATELNÉ ČÍTAČE*
- Obsahují po třech šestnáctibitových čítačích s předvolbou
- Lze jím: Generovat *CLK*
- Vytvořit číslicový *MSKO*
- Řídit krokový motor
- Odměřovat reálný čas

Odměřovat nereálné časy vetřelců z paralelní dimenze
Posekat zahradu

12a) Jazyk OS a jeho interpretace

- o-program je složen z *kroků* (volání a provedení výkonného prog.)
- činnost o počítače je složena z *řídících příkazů* a *výkonných* programů
- příkazy výkonných programů nelze používat jako řídící příkazy vice versa a proto je dobré rozdělit o-jazyk na ro-jazyk a vo-jazyk
- *ro-jazykem* je vyjádřeno jaké výkonné programy mají být spuštěny, s jakými daty, v jakém pořadí a za jakých podmínek
- činnost počítače tedy sestává z řízení a výkonů, lze tedy o-počítač rozdělit na ro-počítač a vo-počítač a může jich běžet několik zároveň
- ro-jazyk neumožňuje větvení a cykly
- vo-jazyk lze zaměnit s k-jazykem, protože jsou prakticky stejné, ale ve vo-jazyku nelze v zájmu bezpečnosti vykonávat privilegované instrukce k-jazyka -> vo-jazyk se na úrovni k neinterpretuje

12b) 8255A

- *PARALELNÍ VSTUPNÍ/VÝSTUPNÍ BRÁNY*
- Obsahuje tři brány a připojují se skrze něj k procesoru třeba periférie nebo topinkovače
- Má tři mody:
- *MODUS 0*: Každá brána funguje nezávisle nastavena jako vstupní nebo výstupní bez zajištění korespondenčního režimu
- *MODUS 1*: Umožňuje vstup nebo výstup branou A a B v korespondenčním režimu. Každá z těchto bran má přiděleny také 3 bity z brány C, které používá pro signály: IBF - Input buffer full, OBF - Output buffer full nebo třeba INTR - Interrupt request
- *MODUS 2*: Funguje pouze brána A, která ale může díky logice řízení fungovat jako vstupní i výstupní

13a) Požadavky na strojový jazyk p. s OS

- je třeba rozlišovat, kdy počítač interpretuje instrukce k-jazyka a kdy vo-jazyku, v zájmu funkčnosti a bezpečnosti
- jsou tedy dva druhy k-instrukcí - *privilegované* a *neprivilegované*; neprivilegované může používat vo-počítač v režimu program; privilegované instrukce může používat systém (k-počítač) v režimu supervizor; toto umožňuje vyhradit některé funkce pouze operačnímu systému a uživatel si je může požadovat skrze službu
- taková služba provede *přerušování*, přepne k-procesor do režimu *supervizor* a předá řízení k-počítači, který vykoná požadovaný k-program
- říci něco o přerušování a o výjimkách

13b) Druhy skoků 8086

- Skákat lze na *návěští*
- Návěští je buď *blízké (NEAR)* nebo *vzdálené (FAR)*; NEAR je pouze offsetem adresy v daném segmentu, zatímco FAR se musí specifikovat direktivou *LABEL* a je úplnou adresou
- Podle druhu návěští rozlišujeme daleké a blízké skoky
- Daleký skok je nepodmíněný skok JMP a jeho podmínění je nutno zařídit jinak
- Blízké skoky jsou všechny skoky podmíněné
- Skok může být podmíněn obsahem flagů

14a) Prostředky a vnější zařízení o-p. a jejich ovládání

- jedna možnost je, že někdy pracuje pouze vnější zařízení a někdy pouze procesor, což je neefektivní

- synchronizaci procesoru s vnějším zařízením za účelem maximální efektivity lze dosáhnout odesláním příkazu k zahájení práce; po tomto odeslání jsou tři možnosti:

A) Vypočítá, jak dlouho bude trvat vnějším zařízením práce a mezitím pracovat

B) Vnější zařízení nastavuje určitý registr po ukončení činnosti; procesor po tuto dobu čeká v čekací smyčce a nic nedělá

C) Vnější zařízení žádá o přerušování, které je okamžitě vykonáno; V tomto přerušování může být obslužný program a lze v něj zadat i novou práci.

- každá úroveň počítače má své prostředky, čímž je myšlen procesor, hlavní paměť a vnější zařízení

- z pohledu uživatele na úrovni n se n-prostředky nazývají virtuální prostředky a n-1-prostředky jako reálné prostředky

- o-procesor je realizován k-procesorem; o-procesorů může být více, potom je několika o-procesorům střídavě odebírán a přidělován k-procesor

- k-prostředky lze rozdělit na pevně přidělené a na odnímatelné; odnímatelné prostředky lze o-prostředkům odnímat libovolně, ale pevně přidělené jen za spolupráce o-programu

- některé k-prostředky jako k-paměť mohou být přidělovány o-prostředkům částečně

- o-periferie sestává z pevně přidělené k-periferie a obslužného programu pro tuto k-periferii; jedna o-periferie může využívat více k-periferií

- příklad vztahu o- a k-periferie je soubor(o-) a svazek(k-)

14b) Způsoby adresování u 8086

- Adresa je složena ze dvou částí - *segmentu* a *offsetu*, přičemž segmentová adresa je uložena v segmentovém registru

- Offset si můžeme vytvořit několika způsoby adresování:

- Adresování: Adresování přímého operandu - MOV AX, 4

Přímé adresování - MOV AX, [4]

Nepřímá adresace - MOV AX, [[4]]

Adresování s registrem - MOV AX, [BX+SI+4]

Adresování s indexem - MOV AX, [4+SI]

15a) Procesor a paměť o-p.

- hlavní paměť k počítače má jednu svou část trvale přidělenou operačnímu systému

- o-paměť má většinou stejně velká paměťová místa jako k-paměť, pomocí níž je implementována, ale umístění o-paměti v k-paměti se může dynamicky měnit s během o-programu

- u *monoprogramních* OS je implementace o-paměti jednoduchá -> je to právě ta k-paměť, která není přidělená operačnímu systému a způsob adresace o-paměti bude vyplývat ze způsobu adresace k-paměti

- u *sdílených pamětí* se využívá *dynamického překládání adres - segmentace*; paměť přidělována programům je rozdělena na segmenty, které mohou ležet na libovolném místě v paměti a mají vlastní registr umístění segmentu, registr délky segmentu; celá adresa je pak určena segmentovým registrem a adresou relativní k začátku segmentu

- o-procesor má o-registry, které se některé shodují s k-registry a

některé k-registry jsou přístupné pouze v režimu supervizor
- o-processoru je přidělován a odjímán k-processor; k-processor pracuje v režimu program, provádí-li instrukce o-programu a v režimu supervizor, interpretuje-li je
- k-processor lze vždy přidělit pouze jednomu o-processoru a ostatní čekají; přidělování může být postupné, podle priority, nebo podle časovače - ten po přesně určených časových okamžicích žádá o přerušeni, aby mohly všechny o-processory využívat k-processor stejnoměrně

15b) Assembler, syntaxe, segmenty, paměťové modely

- *Assembler* je strojovým jazykem a-úrovně počítače, tedy z dnešního pohledu úrovně velmi nízké
- Je kompilovaným jazykem a k jeho kompilaci se využívá kompilátorů různých firem, či volně přístupných kompilátorů, například NASM - NetWide Assembler nebo TASM - Borland Turbo Assembler
- Dříve se Assembleru říkalo jazyk symbolických adres, protože nepracuje přímo s adresami, ale se symboly, které je představují
- Programový kód sestává z *direktiv* a *instrukcí*; direktiva řídí činnost překladače a instrukce je symbolickým vyjádřením instrukce strojového kódu
- Na jedné řádce jedna instrukce ve tvaru: Návěští: Instrukce Cíl, Zdroj ;komentář
- Návěští je buď blízké (NEAR) nebo vzdálené (FAR); NEAR je pouze offsetem adresy v daném segmentu, zatímco FAR se musí specifikovat direktivou *LABEL* a je úplnou adresou
- Cíl a Zdroj jsou operandy, komentář oddělen středníkem
- Program je výhodné rozdělit do segmentů, jejichž velikost a uspořádání určujeme prostřednictvím paměťového modelu, direktivou *MODEL* - segmenty kódu a dat pak budem odlišovat direktivami *CODESEG* a *DATASEG*, zásobník se inicializuje direktivou *STARTUPCODE*, šablona programu pak vypadá nějak takto:

```
IDEAL  
MODEL TINY
```

```
DATASEG  
;data  
CODESEG  
STARTUPCODE  
;kód  
EXITCODE  
END
```

- *EXITCODE* správně ukončuje program a předává řízení OS

16a) Charakteristika MS-DOS

- DOS se skládá ze: Základu systému
Služebních programů
Překladačů
- Základ systému tvoří zaváděcí program, který nahraje jádro operačního systému do paměti, dále vazební program *BIO.COM*, který zprostředkovává styk s hardwarem, dále jádro DOSu - *DOS.COM*, které se stará o řízení IO operací, správu paměti, disků apod.
- *COMMAND.COM* je shell, který zprostředkovává styk s uživatelem a má tři části - rezidentní, která je stále v paměti, inicializační, která se používá při zavádění systému a tranzientní, která slouží

k vlastnímu provádění příkazů

- Služební programy jsou programy i třetích stran, které zjednodušují práci
- Překladače se starají o kompilaci nebo interpretaci kódu napsaném v některém vyšším programovacím jazyce
- Zmínit se o adresářové struktury

16b) Organizace paměti u p. s procesorem 8086 a způsoby adresace

- Program v Assembleru má v paměti vlastní datový segment a segment kódu

- V segmentu kódu je uložen kód programu samotný a v datovém segmentu jsou uložena data nebo paměťová místa, která na data čekají

- Paměťový záznam v datovém segmentu vytvoříme následující direktivou: jméno direktiva (n) (DUP) obsah, kde jméno je symbolická adresa paměťového místa, DUP je direktiva duplikace, n je počet duplikací a obsah je obsahem paměťového místa; direktiva určuje velikost paměťového místa a může to být jedna z těchto:

- db - 1 byte - 8 bitů
- dw - 1 slovo - 16 bitů
- dd - 1 dvojslovo - 32 bitů
- dq - 1 čtyřslovo - 64 bitů
- fd - 6 bytů - FAR ukazatel
- dt - 10 bytů - pro reálná čísla

17a) CP/M a Unix

- CP/M byl před DOSem; Microsoft koupil jeho zdrojové kódy
- CP/M vznikl původně pro p. s procesory 8080, ale byl konstruován jako přenosný, takže se dostal i na jiné platformy - mj. i 8086
- CP/M má prakticky stejnou adresářovou strukturu a strukturu příkazů shellu jako DOS
- UNIX je typ systému, jehož jádro bylo napsáno víceméně celé v programovacím jazyku C a je vrstvené
- První vrstvou je jádro samo, druhou vrstvou jsou knihovny a systémová volání a třetí vrstvou jsou systémové programy
- Jádro sestává z:
 - Systemových rutin
 - Systemových záznamů
 - Systemové vyrovnávací paměti
 - Interních rutin jádra
- Využívá principu sdílení času - přiděluje procesor pravidelně jednotlivým procesům
- Využívá principu swapování - po určitém čase odsouvá z operační paměti procesy na swapovací prostor disku, který je řízen procesorem a je uživateli nepřístupný
- Zmínit se o klonech: Linux (zde uvést, že existuje n distribucí), FreeBSD, OpenBSD, Solaris

17b) Základní typy dat v Assembleru, číselné soustavy

- Paměťový záznam v datovém segmentu vytvoříme následující direktivou: jméno direktiva (n) (DUP) obsah, kde jméno je symbolická adresa paměťového místa, DUP je direktiva duplikace, n je počet duplikací a obsah je obsahem paměťového místa; direktiva určuje velikost paměťového místa a může to být jedna z těchto:

- db - 1 byte - 8 bitů
- dw - 1 slovo - 16 bitů
- dd - 1 dvojslovo - 32 bitů
- dq - 1 čtyřslovo - 64 bitů
- fd - 6 bytů - FAR ukazatel
- dt - 10 bytů - pro reálná čísla

- Další komplexnější datové typy jsou:
- Výčtové - ENUM; struktury - STRUC; uniony - UNION; bitová pole - RECORD; tabulky - TABLE, pojmenovaný datový typ - TYPEDEF

18a) Počítačové sítě, základní pojmy

- Počítačová síť sestává z dvou nebo více počítačů, vzájemně propojených pomocí *systemu dálkového přenosu informací*
- SDPI může být vytvořen veřenou telefonní sítí, pomocí přenosových okruhů, telekomunikačních řídicích jednotek a nebo speciálních procesorů
- Komunikace uživatele se sítí je zprostředkována pomocí *terminálů*, kterým může být i osobní počítač
- Hardware, který zprostředkovává síť: Modemový adaptér
Programovatelný multiplexer
Koncentrátor
Uzlový počítač
- Počítačové sítě mohou být lokálního charakteru - LAN, mezipodnikového či metropolitního charakteru - MAN, osobního charakteru - PAN a také širokého charakteru - WAN
- Nějaký kecy domyslet

18b) Strukturované typy dat v Assembleru

- *STRUKTURA* - obsahuje jednu nebo více proměnných, které sdružuje do jednoho logického celku; velikost struktury je dána součte všech jejích prvků a její definice se uvozuje direktivami STUC a ENDS
- *UNION* - je datová struktura, která umožňuje různý přístup k jednomu paměťovému místu; velikost uniony je dána nejdelším jejím prvkem, protože prvky teoreticky leží přes sebe; například k ukazateli segment:offset můžeme skrze union přistupovat jako k segmentu zároveň a k offsetu zároveň; uvozuje se direktivami UNION a ENDS
- *BITOVÉ POLE* - je datová struktura, jejíž položky nejsou určeny bajty, ale bity; určuje se direktivou RECORD
- *TABULKA* - direktiva TABLE - představuje tabulku hodnot, jde tedy o dvourozměrné pole - tenzor druhého řádu

19a) Druhy počítačových sítí (LAN, WAN, PAN, MAN, něco o TCP/IP)

- *LAN* - Local area network - Místní síť v rámci místnosti nebo budovy
- *WAN* - Wide area network - Širší síť v rámci lokality
- *MAN* - Metropolitní - Většinou podniková síť v rámci města či kraje
- *PAN* - Personal area network - Osobní síť
- Sítě mohou být uzavřené (tj. nepřístupné veřejnosti - příkladem nechť je vojenská síť) nebo otevřené (tj. přístupné veřejnosti - příkladem nechť je Internet)
- Sítě mohou být bezdrátové (tj. propojené fyzicky pomocí elektrického pole ve vodiči nebo optickým kabelem) nebo bezdrátové (tj. propojené pomocí rádiových vln - např. WIFI - wide fidelity nebo technologie Bluetooth či satelitní přenos)

19b) Služby BIOSu a DOSu u PC a jejich volání

- Služby jsou interní přerušování
- V AH je druh služby, v ostatních registrech potom různé parametry určené službě
- *Služby BIOSu*: Služby pro monitor - INT 0x10 (0- nastav videomód, 1- velikost kurzoru, 2- pozici kurzoru, 3- vrať pozici a velikost kurzoru, 5- vyber aktivní videostránku, 6,7- roluj okno, 8- čti znak a atributy z pozice kurzoru, 9- zapiš znak a atributy na pozici

kurzoru ... atp.)

Služby pro klávesnici - INT 0x16 (0- vrať klávesu z bufferu, 1- testuj stav bufferu, 2- vrať stav přepínacích kláves, 5- vlož znak do klávesnicového bufferu)

- *Služby DOSu*: 1- čtení znaku s opisem, 2- zápis znaku, 3- čtení ze sériové linky, 4- zápis do sériové linky, 5- zápis salbiky na tiskárnu, 6- přímý IO zařízení CON, 7- čtení znaku bez opisu, 9,10- zápis, čtení řetězce znaků ... atp.

20a) Topologie počítačových sítí

- *LAN* - Local area network - Místní síť v rámci místnosti nebo budovy

- *WAN* - Wide area network - Širší síť v rámci lokality

- *MAN* - Metropolitní - Většinou podniková síť v rámci města či kraje

- *PAN* - Personal area network - Osobní síť

- *Sít'ová technologie: RING* - Vysílací část předchozího uzlu jest zapojena do přijímací části uzlu následujícího. Provoz je jednosměrný, vhodné použít optické spojení, časté pády, používá se dvojitý prstenec.

STAR - Počítače jsou spojeny pomocí centrálního uzlu. Vhodné pro ethernet i komunikační ústředny

- *Sítě spojové*: Uzly se musí domluvit s aktivními prvky a vytvořit komunikační kanál

- *Sítě nespojové*: Žádný kanál se nevytváří

- *Přenosová média*: Koaxiální kabely stíněné i nestíněné, UTP kabely, optické kabely

20b) Registry a čítače - rozdělení a vlastnosti

- *Registry*: Jsou elementární paměti o kapacitě n1 až n10 bitů a v základním dělení jsou paralelní nebo sériové; sestávají se z D klopných obvodů (tj vstup D, T; výstup Q, Qneg)

- *Paralelní registry*: T spojena, přivedena na STROBE, D vstupy, Q výstupy

- *Sériové registry*: T spojena, přivedena na STROBE, první D je vstup, výstup Q přiveden na další D a poslední Q je výstup; tomuto registru se říká také posuvný registr, ze kterého lze důmyslně vytvořit kruhový registr

- *Čítače*: Se dělí podle směru čítání - nahoru, dolů, reverzibilní, podle reakce výstupů na vstup - asynchronní (výstup se mění postupně od nejnižšího po nejvyšší řád) a synchronní (výstupy se mění naráz) a podle počtu bitů (jeden bit = jeden KO; počet impulzů $P=2^n$)

- *Asynchronní čítač*: CLK na T, Q na další T, Qčka na bitový výstup

21a) Rozlehlé počítačové sítě

- *WAN* - Wide area network - Širší síť v rámci lokality

- *MAN* - Metropolitní - Většinou podniková síť v rámci města či kraje

- *Internet* - Mezinárodní decentralizovaná počítačová síť

- Internetu předcházela univerzitní síť *ARPANET*, ve které se také objevil historicky první případ *počítačové kriminality*

- Převážné využití decentralizovaných uzlů -> Odolnost proti fyzickému narušení sítě například vojenskými silami

- Nejednoznačné cesty signálů

- V čechách jsou takové sítě např. *CESNET* nebo *BUDNET* v rámci ČB

21b) Základní instrukční typy v Assembleru

- Tady bych se chtěl předem omluvit, protože nepopíši funkci, jelikož ji znám velmi dobře

- Přesunové : MOV, PUSH, POP, XCHG

- Střadačové: XLAT, IN, OUT
- Pro práci s adresami: LEA,
- Pro práci s příznaky: LAHF, SAHF, PUSHF, POPF
- Aritmetické: ADD, ADC, INC, AAA, DAA, SUB, SBB, DEC, NEG, CMP, MUL, IMUL, DIV, IDIV
- Logické: NOT, AND, OR, XOR, TEST
- Posuvů: SHR, SHL, SAR, SAL, ROL, ROR, RCL, RCR
- Pro práci s řetězci: REP, MOVS, CMPS, SCAS, LODS, STOS
- Pro řízení toku:
 - Cykly: LOOP, LOOPNE
 - Skokové: JMP, JNZ a další podmíněné
- Pro práci s podprogramy: CALL, RET
- Vnitřních přerušeni: INT, INTO, IRET
- Pro nastavování příznaků: STn, CLn
- Ostatní: ESC, John LOCKE, NOP

22a) Terminály a terminálové stanice, terminálové sítě

- *Terminálová síť* je výpočetní systém s vícenásobným přístupem, realizovaný spojením počítače s více koncovými zařízeními - terminály
- Pro zabezpečení funkce musí síť splňovat požadavky v těchto třech úrovních: *Technické úrovní* - Terminály, přenosová cesta a server musí mít vhodnou hardwarovou konfiguraci
 - Programové úrovní* - Programové vybavení musí splňovat veškeré nároky a to jak obslužný software, tak operační systém
 - Organizační úrovní* - Organizace práce a pravidla užívání sítě
- Podle vzdálenosti terminálů od serveru se rozlišují *lokální terminálové sítě* a *rozlehlé terminálové sítě*
- Výhody terminálové sítě: Rychlý a pohotový přístup k informačním a výpočetním zdrojům, možnost využití jednoho zdroje více uživateli současně.
- *Terminál* je libovolné zařízení, skrze které se můžeme oboustraně dorozumívat s výpočetním nebo informačním zdrojem

22b) Struktury a uniony v Assembleru

- *STRUKTURA* - obsahuje jednu nebo více proměnných, které sdružuje do jednoho logického celku; velikost struktury je dána součte všech jejích prvků a její definice se uvozuje direktivami STUC a ENDS
- *UNION* - je datová struktura, která umožňuje různý přístup k jednomu paměťovému místu; velikost uniony je dána nejdelším jejím prvkem, protože prvky teoreticky leží přes sebe; například k ukazateli segment:offset můžeme skrze union přistupovat jako k segmentu zároveň a k offsetu zároveň; uvozuje se direktivami UNION a ENDS

23a) Počítačové sítě

- Výběr z toho, co už jsem napsal výše

23b) Bitové pole a výčtové typy v Assembleru

- *BITOVÉ POLE* - je datová struktura, jejíž položky nejsou určeny bajty, ale bity; určuje se direktivou RECORD
- *TABULKA* - direktiva TABLE - představuje tabulku hodnot, jde tedy o dvourozměrné pole - tenzor druhého řádu
- *VÝČTOVÝ TYP* - ENUM - Enumerace - Programátor enumerací definuje vlastní typ strukturované proměnné, která bude obsahovat přesně ty vnitřní proměnné, které enumeroval

24a) Základní propojovací prvky p. sítí

- *Opakovač*: Obousměrný číslicový zesilovač. Zvětšuje dosah sítě
- *Síťový most*: Síťový most spojuje dvě různé lokální sítě lišící se ve fyzické a linkové vrstvě
- *Switch*: Spojuje dvě různé sítě - například LAN a WAN
- *Směrovač*: Mosty, které se starají o hledání optimální cesty
- *HUB*: Host Universal Bus - Univerzální síťová sběrnice
- Sítě se propojují koaxem, UTPčkem, optikou nebo vzduchem, viz 20

24b) Základní znakové kódování v PC a vektory přerušení

- *Znakové kódování* je systém překládání číselných kódů na znaky čtené uživatelem
- *ASCII* - nejpoužívanější znaková sada vůbec, ostatní jí prakticky pouze rozšiřují
- Znakové sady podporující češtinu: *Unicode, Windows-1250, Kód kamenických*
- Na nejnižších adresách paměti se nachází takzvaná *tabulka vektorů přerušení*, skládá se z 256 4 bytových adres, přičemž každá začíná adresou přerušení
- Tabulku přerušení lze dle chuti přepisovat, vytvářet vlastní přerušení a jejich obsluhy

25a) Lokální počítačové sítě a internet

- *LAN* - Local area network - Místní síť v rámci místnosti nebo budovy
- *Internet* - Mezinárodní decentralizovaná počítačová síť
- Internetu předcházela univerzitní síť *ARPANET*, ve které se také objevil historicky první případ počítačové kriminality
- Převážné využití decentralizovaných uzlů -> Odolnost proti fyzickému narušení sítě například vojenskými silami
- Nejednoznačné cesty signálů
- *WEB* - nejpoužívanější služba internetu
- *E-Mail* - druhá nejpoužívanější služba internetu
- *Usenet* - diskusní skupiny; zpočátku stály mimo internet
- *IRC* - internet relay chat, kdysi největší služba pro výměnu zpráv využívající internet
- Podobný žvásty, co každéj normální člověk dokáže sám a nepotřebuju k tomu vodit za ruku Michaelem Flannerym.

25b) Programování obvodu 8255A

- *PARALELNÍ VSTUPNÍ/VÝSTUPNÍ BRÁNY*
- Obsahuje tři brány a připojují se skrze něj k procesoru třeba periferie nebo topinkovače
- Má tři módy:
- *MODUS 0*: Každá brána funguje nezávisle nastavena jako vstupní nebo výstupní bez zajištění korespondenčního režimu
- *MODUS 1*: Umožňuje vstup nebo výstup branou A a B v korespondenčním režimu. Každá z těchto bran má přiděleny také 3 bity z brány C, které používá pro signály: *IBF* - Input buffer full, *OBF* - Output buffer full nebo třeba *INTR* - Interrupt request
- *MODUS 2*: Funguje pouze brána A, která ale může díky logice řízení fungovat jako vstupní i výstupní
- Z toho programování obvodu jsem jelen, takže asi nic nebude :)
- Třeba mu bude stačit, když řeknete, že čipselekt je pořád nula a áčkama se vybírá brána a čímsi jiným modus

ŠLUS! A je to ... Mikroprocesorovka naučena! Zbývá EN a AJ