

<p>Co je správně?</p> <ul style="list-style-type: none"> • Jeden bit má osm bajtů. • Jeden bajt má osm bitů. • Jeden bajt je složen ze dvou nebo čtyř slov. 	<p>Adresový registr obsahuje 4 bity. Kolik je schopen namapovat (zaadresovat) adres?</p> <ul style="list-style-type: none"> • 4 • 8 • 10 • 16 • 20
<p>Nejmenší adresovatelná jednotka paměti je</p> <ul style="list-style-type: none"> • kapacita místa v paměti, které má vlastní adresu. • nejmenší hodnota adresy v paměti. • nejmenší číslo, které lze do paměti uložit. 	<p>Paměť o maximální kapacitě 1 M adresovatelných míst musí mít adresovací sběrnici širokou právě</p> <ul style="list-style-type: none"> • 32 bitů • 21 bitů • 20 bitů • 30 bitů
<p>Nejmenší adresovatelná jednotka paměti typicky je</p> <ul style="list-style-type: none"> • 1 bit • 8 bitů • 16 bitů 	<p>Paměť o maximální kapacitě 1 G adresovatelných míst musí mít adresovací sběrnici širokou právě</p> <ul style="list-style-type: none"> • 32 bitů • 21 bitů • 20 bitů • 30 bitů
<p>1 KB je</p> <ul style="list-style-type: none"> • 1000 B • 1048 b • 1024 B 	<p>Jaká je správná posloupnost seřazená podle velikosti uchovávané informace od nejmenší po největší?</p> <ul style="list-style-type: none"> • bit, slovo, bajt • bit, bajt, slovo • bajt, slovo, bit • bajt, bit, slovo • slovo, bajt, bit
<p>2¹⁰ bajtů je</p> <ul style="list-style-type: none"> • 1 KB • 128 KB • 512 KB • 1 MB 	
<p>2¹⁶ bajtů je</p> <ul style="list-style-type: none"> • 24 KB • 32 KB • 64 KB • 128 KB 	<p>Paměť RAM</p> <ul style="list-style-type: none"> • se řadí mezi paměti se sekvenčním přístupem • je určena pouze ke čtení • je určena ke čtení i k zápisu • se řadí mezi periferní paměti
<p>2²⁰ bajtů je</p> <ul style="list-style-type: none"> • 256 KB • 512 KB • 1 MB • 2 MB • 4 MB 	<p>Doslovný překlad zkratky RAM je</p> <ul style="list-style-type: none"> • Rewrite And Machine • Random Access Memory • Record Access Memory
<p>2³² bajtů je</p> <ul style="list-style-type: none"> • 2 MB • 4 MB • 1 GB • 2 GB • 4 GB 	<p>Vestavěný program řídící činnost automatického jednoúčelového zařízení patří typicky do kategorie</p> <ul style="list-style-type: none"> • hardware • bestware • firmware • adware • spyware

Jednotka informace 1 slovo (1 word) odpovídá

- 80 b
- 2 B
- 32 b
- 64 b
- **všechny odpovědi mohou být správně**

Jedno slovo obvykle nemá

- **1 slabiku**
- 2 slabiky
- 4 slabiky
- 8 slabik

Kontrolní bit například na děrné pásce se nazývá

- párový bit
- partikulární bit
- paralelní bit
- parciální bit
- **paritní bit**

24bitová adresová sběrnice dokáže adresovat paměťový prostor o kapacitě maximálně (adresovatelná jednotka je bajt):

- 4 MB
- **16 MB**
- 1 GB
- 4 GB
- 16 GB

Mezi různými typy pamětí nejmenší kapacitu má obvykle

- **registr**
- vnitřní (operační) paměť
- vnější (periferní) paměť

Mezi různými typy pamětí je z hlediska přístupu nejrychlejší paměť

- **registr**
- vnitřní (operační) paměť
- vnější (periferní) paměť

Paměť se sekvenčním přístupem

- má vždy kratší přístupovou dobu k datům než paměť s přímým přístupem
- **při přístupu k místu s adresou n projde nejdříve adresy 0-(n-1)**
- je typicky paměť typu registr
- je typicky vnitřní (operační) paměť

Která charakteristika neplatí pro paměť typu registr?

- velmi malá kapacita
- **energeticky nezávislá**
- velmi nízká přístupová doba
- paměť s přímým přístupem
- slouží pro krátkodobé uchování právě zpracovávaných informací

Která charakteristika platí pro paměť typu registr?

- kapacita v řádu desítek GB
- energeticky nezávislá
- **paměť s přímým přístupem**
- slouží pro dlouhodobé uchování informací
- při přístupu k místu s adresou n projde nejdříve adresy 0-(n-1)

Architektura počítače "von Neumann" obsahuje pravidlo:

- Počítač obsahuje procesor, DMA kanál, operační paměť a V/V zařízení.
- **Počítač obsahuje operační paměť, ALJ, řadič a V/V zařízení.**
- Počítač obsahuje procesor, DMA kanál a operační paměť.

Architektura počítače "von Neumann" obsahuje pravidlo:

- **Údaje a instrukce jsou vyjádřeny binárně.**
- Údaje a instrukce jsou vyjádřeny číselně.
- Údaje a instrukce jsou vyjádřeny slovně.
- Instrukce se v assembleru píše zkratkou.

V architektuře "von Neumann" má dekódování instrukcí na starost

- **řadič**
- aritmeticko-logická jednotka
- procesor
- operační paměť
- V/V zařízení

Které tvrzení neplatí pro von Neumannovu architekturu?

- **Program je uložen v paměti oddělené od paměti pro data.**
- Počítač obsahuje operační paměť, ALJ, řadič a V/V zařízení.
- Program je uložen v paměti spolu s daty.
- Instrukce jsou vyjádřeny binárně.
- Data jsou vyjádřena binárně.

Stavová hlášení jsou v architektuře "von Neumann" zasílána:

- aritmeticko-logické jednotce
- operační paměti
- **řadiči**
- V/V zařízení
- procesoru

Které tvrzení o koncepci Johna von Neumanna neplatí?

- Program se umístí do operační paměti přes ALJ pomocí vstupního zařízení.
- Data se umístí do operační paměti přes ALJ pomocí vstupního zařízení.
- Jednotlivé kroky výpočtu provádí aritmeticko-logická jednotka.
- Mezivýsledky jsou ukládány do operační paměti.
- **Po skončení jsou výsledky posílány přes řadič na výstupní zařízení.**

Ve von Neumannově modelu

- netečou data z ALJ do paměti
- **netečou data z řadiče do ALJ**
- netečou data z ALJ do řadiče
- netečou data z paměti do ALJ

Mezi typickou činností řadiče patří

- **transformuje instrukce na posloupnost signálů ovládající připojené zařízení**
- poskytuje paměťový prostor pro data, která tečou do procesoru
- slouží jako podpůrná výpočetní jednotka pro ALJ
- transformuje logickou adresu na fyzickou

DMA je určeno především pro

- ukládání často užívaných instrukcí
- **přenos dat z disku do operační paměti**
- korekci obrazového výstupu
- kontrolu dat ukládaných na disk
- provádění aritmetických operací

V polyadické soustavě je číslo

- součet bitů n-tice, ve které je uloženo.
- vždy dělitelné svým základem.
- **součet mocnin základu vynásobených číslicemi.**

Číslo lze snadno (každou k-tici číslic nižší soustavy nahradíme číslicí soustavy vyšší) převádět mezi soustavami o základu

- 5 a 7
- **8 a 2**
- 10 a 16

Číslo 21 v desítkové soustavě po převedení do soustavy dvojkové je

- **10101**
- 11011
- 10011
- nelze do dvojkové soustavy převést

Pascalovský typ INTEGER je celé číslo, které se na počítačích PC zobrazuje v

- přímém kódu.
- **doplňkovém kódu.**
- inverzním kódu.

Znaménkový bit v celém čísle je zpravidla bit

- nejnižšího řádu.
- nultého řádu.
- **nejvyššího řádu.**

Znaménkový bit bývá zpravidla

- roven jedné, pokud se zobrazuje číslo kladné
- roven nule, pokud se zobrazuje číslo záporné
- **roven nule, pokud se zobrazuje číslo kladné**

Rozsah zobrazení celého čísla uloženého ve dvojkovém doplňkovém kódu na 8 (celkem) bitech je

- **<-128;127>**

- $\langle -256; 255 \rangle$
- $\langle -511; 512 \rangle$
- $\langle -1024; 1023 \rangle$
- žádný z uvedených

Největší zobrazitelné celé číslo ve dvojkovém doplňkovém kódu má tvar

- 100...00
- 111...11
- 000...00
- 100...01
- **011...11**

Při sčítání dvou čísel v inverzním kódu jako korekci výsledku použijeme:

- násobný přenos
- **kruhový přenos**
- konverzní přenos
- desítkový přenos

Přeplnění (přetečení) je stav, ve kterém

- výsledek spadá mimo přesnost
- výsledek spadá mimo rozlišitelnost
- **výsledek spadá mimo rozsah zobrazení**

Vyberte nepravdivé tvrzení týkající se zobrazení celého čísla:

- přímý kód obsahuje kladnou a zápornou nulu
- inverzní kód obsahuje kladnou a zápornou nulu
- doplňkový kód obsahuje pouze jednu nulu
- **rozsah zobrazení doplňkového kódu je symetrický**
- se všemi bity doplňkového kódu se pracuje stejně

Inverzní kód pro zobrazení celého čísla nemá

- **jednu nulu**
- symetrický rozsah zobrazení
- znaménkový bit
- ve znaménkovém bitu jedničku pro označení záporného čísla

Znaménkový bit pro zobrazení celého čísla

- je bit nejnižšího řádu
- se běžně nepoužívá
- je bit nejnižšího řádu pouze pokud se jedná o číslo
- má hodnotu 1 pro kladné číslo
- **má hodnotu 0 pro kladné číslo**

Přetečení v celočíselné aritmetice ve dvojkovém doplňkovém kódu nastane

- pokud se přenos ze znaménkového bitu rovná přenosu do znaménkového bitu
- **pokud se přenos ze znaménkového bitu nerovná přenosu do znaménkového bitu**
- pokud se přenos ze znaménkového bitu nerovná znaménkovému bitu
- pokud se přenos ze znaménkového bitu rovná znaménkovému bitu
- pokud výsledek operace nespadá mimo rozsah zobrazení

Osmičkovou a šestnáctkovou soustavu používáme, protože:

- vnitřně si počítač uchovává data v těchto soustavách
- výpočet procesoru je rychlejší než při použití dvojkové soustavy
- **zápis čísla je kratší než ve dvojkové soustavě**
- vstupní a výstupní zařízení pracují s těmito soustavami

Binární hodnota 0,1001 odpovídá dekadické hodnotě desetinného čísla:

- **9/16**
- 1/32
- 9/10
- 1/16
- 10/9

Při sčítání ve dvojkovém doplňkovém kódu platí:

- přetečení nastane, pokud je rozsah zobrazení jiný než $\langle 0; 2^n - 1 \rangle$
- všechny bity (kromě znaménkového) se sčítají stejně
- vznikne-li přenos ze znaménkového bitu, je nutné provádět tzv. kruhový přenos
- přetečení nastane, pokud se přenosy z/do znaménkového bitu rovnají
- **vznikne-li přenos ze znaménkového bitu, tak se ignoruje**

Dvojkové číslo 1000 v přímém kódu v zobrazení se znaménkem na 4 bitech je: **Kladná čísla v reprezentaci bez znaménka mají na n bitech rozsah:**

- největší zobrazitelné
 - nejmenší zobrazitelné
 - kladná nula
 - **záporná nula**
 - žádná odpověď není správná
- **$\langle 0; 2^n - 1 \rangle$**
 - $\langle 0; 2^{n-1} - 1 \rangle$
 - $\langle 0; 2^{n-1} + 1 \rangle$
 - $\langle -2^{n-1}; 2^n - 1 \rangle$
 - $\langle -2^{n-1} - 1; 2^{n-1} - 1 \rangle$

Dvojkové číslo 1000 v inverzním kódu v zobrazení se znaménkem na 4 bitech je:

- největší zobrazitelné
- **nejmenší zobrazitelné**
- kladná nula
- záporná nula
- žádná odpověď není správná

Dvojkové číslo 1111 v doplňkovém kódu v zobrazení se znaménkem na 4 bitech je:

- největší zobrazitelné
- nejmenší zobrazitelné
- kladná nula
- záporná nula
- **žádná odpověď není správná**

Kruhový přenos je:

- inverze bitů
- inverze bitů a přičtení jedničky k výsledku
- **přičtení přenosu z nejvyššího řádu k výsledku**
- přičtení přenosu z nejvyššího řádu ke znaménkovému bitu
- přičtení jedničky k nejvyššímu řádu výsledku

Kladná čísla v zobrazení se znaménkem mají na n bitech:

- **ve všech kódech stejný rozsah**
- v přímém kódu o 1 větší rozsah než v inverzním
- stejný rozsah jako kladná čísla v zobrazení bez znaménka
- v inverzním kódu o 1 číslo méně, než je záporných
- v inverzním kódu rozsah $\langle 0; 2^n - 1 \rangle$

Které z dvojkových čísel v reprezentaci se znaménkem na 4 bitech je kladné?

- 1010 v inverzním kódu
- **0100 v inverzním kódu**
- 1010 v přímém kódu
- 1111 v doplňkovém kódu
- všechny odpovědi jsou správné

Rozsah zobrazení směrem ke kladným číslům a směrem k záporným číslům je rozložen asymetricky v:

- přímém kódu
- inverzním kódu
- **doplňkovém kódu**
- přímém a inverzním kódu
- inverzním a doplňkovém kódu

Dvě reprezentace nuly se vyskytují v:

- přímém a doplňkovém kódu
- **přímém a inverzním kódu**
- inverzním a doplňkovém kódu
- doplňkovém, inverzním a přímém kódu

Která z čísel jsou shodná (nejvyšší bit je znaménkový)?

- 1001 v přímém a 1010 v inverzním kódu
- **1101 v inverzním a 1110 v doplňkovém kódu**
- 1111 v doplňkovém a 1000 v přímém kódu
- 1000 v doplňkovém a 1000 v inverzním kódu
- žádná z odpovědí není správná

Rozsah zobrazení dvojkového doplňkového kódu na n bitech je:

- $\langle 0; 2^{n-1} - 1 \rangle$
- **$\langle -2^{n-1}; 2^{n-1} - 1 \rangle$**
- $\langle -2^{n-1} - 1; 2^{n-1} - 1 \rangle$
- $\langle -2^{n-1}; 2^n - 1 \rangle$
- $\langle -2^n + 1; 2^n - 1 \rangle$

Dvojkové číslo 1001 v reprezentaci se znaménkem na 4 bitech se v inverzním kódu rovná

- 6
- **-6**
- 9
- -9

Jak při sčítání binárních čísel ve dvojkovém doplňkovém kódu poznám, že došlo k přetečení?

- k přetečení nemůže dojít, zabraňuje mu kruhový přenos
- přenos ze znaménkového bitu je 1
- **přenos do znaménkového bitu se nerovná přenosu ze znaménkového bitu**
- přenos do znaménkového bitu se rovná přenosu ze znaménkového bitu

Číslo 14 v decimální soustavě odpovídá

- D v hexadecimální soustavě
- 15 v oktalové soustavě
- 1101 v binární soustavě
- **E v hexadecimální soustavě**

Kruhový přenos v inverzním kódu se využívá

- **pro korekci při přechodu přes nulu**
- pro zkopírování nejnižšího bitu do nejvyššího
- pro zkopírování nejvyššího bitu do nejnižšího
- kruhový přenos se v inverzním kódu nepoužívá

Jednoduše nelze převádět čísla mezi soustavami o základech

- 5 a 25
- 3 a 9
- **4 a 40**
- 6 a 216
- 6 a 36

Osmičková soustava se také nazývá

- oktetová
- **oktalová**
- oktanová
- oktarová
- oklotová

V ASCII kódu má

- **ordinální hodnota znaku návrat vozíku (CR) menší hodnotu než ordinální hodnota znaku 'A'.**
- ordinální hodnota znaku návrat vozíku (CR) větší hodnotu než ordinální hodnota znaku 'A'.
- znak návrat vozíku (CR) v ASCII kódu vůbec není.

V ASCII kódu jsou znaky s ordinální hodnotou 0 až 31 označeny jako

- **řídící znaky**
- alfanumerické znaky
- alfabetské znaky
- tisknutelné znaky

Písmena s diakritikou nejsou součástí vnějšího kódování

- **ASCII**
- ISO-8859-2
- Windows-1250

Jaké kódování je korektní pro zobrazení všech českých znaků s diakritikou

- ASCII
- ISO-8859-1
- **ISO-8859-2**

Znak "Line feed"

- **je řídící znak s ordinální hodnotou nižší než 30**
- je řídící znak s ordinální hodnotou vyšší než 30
- se nevyskytuje v kódování ASCII-7
- není řídící znak

Řídící znak "Carriage return" znamená

- **přesun na začátek téhož řádku**
- přesun na začátek dalšího řádku
- začátek příkazové řídící sekvence
- přesun na začátek předchozího řádku
- takový řídící znak neexistuje

Pro označení konce řádku v textovém souboru MS-Windows slouží kombinace znaků:

- CR+NUL
- **CR+LF**
- BS+CR
- LF
- CR+DEL

Unicode je

- **vnější kódování znaků**
- sjednocené kódování celých čísel
- způsob ukládání reálných čísel

UTF-8 zobrazuje jeden znak

- vždy jedním bajtem
- vždy dvěma bajty
- **různým počtem bajtů**

Unicode je

- způsob uložení a UTF-8 je vnější kódování
- **vnější kódování a UTF-8 je způsob uložení**

Česká písmena s diakritikou jsou v UTF-8 uložena nejvíce na

- jednom bajtu
- **dvou bajtech**
- třech bajtech
- čtyřech bajtech

UTF-8 uloží znak z ASCII 7 na

- **1 bajtu**
- 2 bajty
- 3 bajty
- 4 bajty
- 5 bajtů

Počet bajtů, v kolika je uložen znak v UTF-8 (je-li uložen ve více než jednom bajtu), je vyjádřen

- **počtem binárních jedniček v bitech nejvyšších řádů**
- počtem binárních nul v bitech nejvyšších řádů
- číslem 0-7 v nejvyšších třech bitech
- číslem 0-7 v nejnižších třech bitech

Vnější kódy ISO-8859-2 a Windows-1250 se liší v ordinální hodnotě znaku

- ñ
- č
- **š**

Detekční kód je kód, který

- nahlásí chybu v počítači.
- **rozpozná chybu v uložené či přenášené informaci.**
- detekuje hackera v počítači.

Opravný kód je kód, který

- najde chybu v systému Windows a opraví ji.
- opraví chybu programátora v jeho zdrojovém kódu.
- **opraví chybu v uložené či přenášené informaci.**

Hammingova trojrozměrná krychle má

- **6 stěn.**
- 2 stěny.
- žádnou stěnu.
- 8 stěn.

BCD (Binary Coded Decimal) znamená

- binárně zakódovaná čísla tak, aby je nešlo dešifrovat.
- desítkově kódovaná binární čísla.
- **jedna desítková číslice uložena vždy na čtyřech bitech.**

BCD znamená

- **Binary Coded Decimal**
- Binary Crowded Decimal
- Binary Coded Hexadecimal
- Bipolar Coded Decimal

BCD kód v každé

- trojici bitů ukládá jednu oktalovou číslici
- čtveřici bitů ukládá jednu šestnáctkovou číslici
- **čtveřici bitů ukládá jednu desítkovou číslici**
- trojici bitů ukládá jednu desítkovou číslici

Kladné číslo v rozvinutém BCD tvaru je

- 71346C
- 71346D
- **F7F1F3F4C6**
- F7F1F3F4F6C
- +F7F1F3F4F6D

Číslo, které je v rozvinutém BCD tvaru uloženo na 5 bajtech, bude ve zhuštěném BCD tvaru uloženo ve

- 2 bajtech
- **3 bajtech**
- 4 bajtech
- 5 bajtech
- 6 bajtech

V čem je uznávaná výhoda zobrazení čísel v BCD kódu oproti zobrazení čísel v přímém binárním kódu?

- **jednodušší převod čísla do desítkové soustavy**
- jednodušší provádění aritmetických operací
- kratší zápis čísla
- BCD kód je nyní všeobecně používanější

Co znamená kód 2 z 5?

- způsob zabezpečení informace, právě dva bity jsou rovny nule
- způsob kódování podobný kódu CP1250
- **způsob zabezpečení, právě dva bity jsou rovny jedné**
- způsob kódování na principu UTF-16

Při Hammingově vzdálenosti (d) pět

- mohu kód opravit, pokud vznikne maximálně jedna chyba
- **mohu kód opravit, pokud vzniknou maximálně dvě chyby**
- mohu kód opravit, pokud vzniknou maximálně tři chyby
- nejsem schopen opravit chybu

Při Hammingově vzdálenosti (d) dva

- **jsem schopen detekovat chybu a nejsem schopen ji opravit**
- jsem schopen detekovat chybu a jsem schopen ji opravit
- nejsem schopen detekovat chybu

Sudá parita znamená

- **počet bitů vč. paritního obsahujících hodnotu 1 je sudý**
- počet bitů vč. paritního obsahujících hodnotu 1 je lichý
- počet bitů bez paritního obsahujících hodnotu 1 je sudý
- počet bitů bez paritního obsahujících hodnotu 1 je lichý
- počet chyb, které jsme schopni detekovat, je sudý

Mějme detekční kód 2 z 5. Které z následujících čísel obsahuje chybu?

- 00101
- **11010**
- 10001
- 00011
- 01100

Ztrojení

- je příkladem vnějšího kódu
- **je příkladem opravného kódu**
- uloží hodnotu tří bitů na jeden bit
- umožňuje detekovat 3 chyby, ale pouze 2 opravit

Kódová (Hammingova) vzdálenost je:

- **počet bitů, v nichž se liší dvě sousední platné kódové kombinace**
- počet bitů, v nichž se shodují dvě sousední platné kódové kombinace
- počet jedničkových bitů ve dvou sousedních platných kódových kombinacích
- počet chyb, které jsme schopni detekovat
- počet chyb, které jsme schopni opravit

Pro Hammingovu vzdálenost 1 platí

- **žádnou chybu nelze detekovat, tedy ani opravit**
- jednu chybu lze detekovat, ale nelze ji opravit
- jednu chybu lze detekovat a je možné ji opravit
- dvě chyby lze detekovat a jednu chybu lze opravit

Kolik chyb jsme schopni detekovat, jestliže kódová vzdálenost $d=3$?

- žádnou
- jednu
- **dvě**
- tři
- čtyři

Kolik chyb jsme schopni opravit, jestliže kódová vzdálenost $d=3$?

- žádnou
- **jednu**
- dvě
- tři
- čtyři

V opravném kódu v případě ztrojení každého bitu

- jsme schopni jednu chybu detekovat a dvě chyby korektně opravit
- jsme schopni jednu chybu detekovat a jednu chybu korektně opravit
- jsme schopni dvě chyby detekovat a obě dvě korektně opravit
- **jsem schopen dvě chyby detekovat a jednu chybu korektně opravit**

Jak **jaké ordinální hodnoty mají číslice v EBCDIC (vnější kód BCD)?** **Sériové zapojení vyjádřené v Booleově algebře znamená**

- A0 až A9
- C0 až C9
- D0 až D9
- E0 až E9
- **F0 až F9**

Co znamená **Big-Endian**

- počítač má jeden konec větší než druhý
- **bajt nejvyššího řádu je na nejnižší adrese**
- bajt nejnižšího řádu je na nejnižší adrese
- bajt nejvyššího řádu je na nejvyšší adrese

Co znamená **použití pořadí Little-Endian?**

- **Bajt nejnižšího řádu je uložen na nejnižší adrese.**
- Bajt nejvyššího řádu je uložen na nejnižší adrese.
- Bajt nejnižšího řádu je uložen na nejvyšší adrese.
- Všechny bity (kromě znaménkového) se sčítají stejně.

Little-Endian a Big-Endian jsou způsoby

- ukládání bitů v bajtu
- **ukládání bajtů ve slově**
- připojování konektorů sběrnice

Jak na čísla ve dvojkovém doplňkovém kódu poznáme, zda je uloženo v Big-Endian nebo Little-Endian

- podle hodnoty nejvyššího bitu
- podle hodnoty nejvyššího bajtu
- podle hodnoty nejnižšího bajtu
- podle hodnoty nejnižšího bitu
- **nelze to ze zápisu čísla jednoznačně poznat**

Mezi operace Booleovy algebry nepatří

- logický součet
- **logický rozdíl**
- logický součin
- negace

- logický součet
- logický rozdíl
- **logický součin**
- negaci

Paralelní zapojení vyjádřené v Booleově algebře znamená

- **logický součet**
- logický rozdíl
- logický součin
- negaci

Který z uvedených způsobů se nepoužívá pro minimalizaci výrazu?

- matematické úpravy
- jednotková krychle
- karnaughova mapa
- **jednotková kružnice**

Proč není Booleova algebra vhodná pro technickou realizaci?

- **obsahuje příliš mnoho operací**
- byla vymyšlena dříve, než se začala uplatňovat von Neumannova koncepce
- zakreslení grafů je pomocí ní příliš obtížné
- není možné pomocí ní provádět operaci implikace

Jaké operace využívá Shefferova algebra?

- **jedinou operaci a to negovaný logický součin (NAND)**
- jedinou operaci a to negovaný logický součet (NOR)
- dvě operace - negovaný logický součin (NAND) a negovaný logický součet (NOR)
- operace logický součin (AND), logický součet (OR) a negaci (NOT)

Shefferova algebra (NAND) se používá místo Booleovy algebry v technických zapojeních, protože

- je rychlejší.
- je levnější.
- **má jen jednu operaci.**
- má více operací.

Zakázané pásmo v obvodech

- je vymezeno nejnižší hodnotou napětí, při které již může dojít k poškození obvodu
- **vymezuje hodnoty signálu, ve kterých se signál nesmí nacházet během jeho vzorkování**
- je maximální vzdálenost mezi dvěma obvody, ve které ještě dochází k nežádoucímu ovlivňování tvaru signálu

Zakázané pásmo v obvodech je

- vzdálenost od počítače, ve které se nesmí vyskytovat jiný spotřebič.
- poloměr kruhu okolo procesoru, ve kterém se nesmí vyskytovat žádný signál.
- **rozsah hodnot, ve kterém se signál nesmí nacházet v okamžiku vzorkování.**

Napájecí napětí technologie TTL je

- **5 V**
- 220 V
- 120 V na americkém kontinentu

Invertor

- je sekvenční logický člen
- je logický člen měnící kladné napětí na záporné
- **je logický člen měnící logickou 0 na logickou 1 a opačně**
- je sekvenční logický člen měnící logickou 0 na logickou 1 a opačně

Výstupní hodnota logického členu NOR je rovna 1, když

- všechny vstupní hodnoty jsou 1.
- aspoň jedna vstupní hodnota je 0.
- aspoň jedna vstupní hodnota je 1.
- **všechny vstupní hodnoty jsou 0.**

Výstupní hodnota logického členu NOR je rovna 0, když

- aspoň jedna vstupní hodnota je 0.
- **aspoň jedna vstupní hodnota je 1.**
- všechny vstupní hodnoty jsou 0.

Výstupní hodnota logického členu NAND je rovna 0, když

- **všechny vstupní hodnoty jsou 1.**
- aspoň jedna vstupní hodnota je 0.
- aspoň jedna vstupní hodnota je 1.
- všechny vstupní hodnoty jsou 0.

Výstupní hodnota logického členu NAND je rovna 1, když

- všechny vstupní hodnoty jsou 1.
- **aspoň jedna vstupní hodnota je 0.**
- aspoň jedna vstupní hodnota je 1.

Mezi kombinační logické obvody patří

- **NAND, NOT, multiplexor**
- RS, JK, AND, OR
- NOR, D, XOR

Mezi kombinační logické obvody patří

- klopný obvod R-S
- **sčítanka pro jeden binární řád**
- jednobitová paměť

Kombinační logický obvod "nonekvivalence" má stejnou funkci jako:

- logický součet
- **sčítanka modulo 2**
- multiplexor

Klopný obvod RS v obecném případě nesmí mít na vstupu kombinaci 00,

- pokud je řízen jedničkami
- **pokud je řízen nulami**
- protože na komplementárních výstupech budou stejné hodnoty

Parita je

- obvod pro vyhodnocení hlasovací funkce.
- způsob porovnání dvou čísel.
- **způsob zabezpečení informace proti chybě.**

Multiplexor se čtyřmi datovými vstupy je obvod, který

- **dle zadané adresy vybere jeden ze vstupních signálů a předá jej na výstup.**
- dle zadané adresy vybere čtyři vstupní signály a sloučí je do jednoho výstupního.
- vybere náhodně jeden ze čtyř vstupních signálů a předá jej na výstup.

Multiplexor se 16 datovými vstupy potřebuje

- **4 adresové vstupy.**
- 16 adresových vstupů.
- 65536 adresových vstupů.

Dekodér, který má 2 vstupy, má

- 2 výstupy.
- **4 výstupy.**
- 8 výstupů.

Úplná sčítačka pro jeden binární řád má

- dva bity sčítanců na vstupu a jeden bit součtu na výstupu.
- dva bity sčítanců na vstupu a jeden bit součtu a přenos na výstupu.
- **dva bity sčítanců a přenos na vstupu a jeden bit součtu a přenos na výstupu.**

Co je pravda?

- **Sekvenční logické obvody mají vnitřní stav.**
- Kombinační logické obvody mají vnitřní stav.
- Nic z toho není pravda.

Zakázaný stav u klopného obvodu R-S řízeného jedničkami je stav, kdy

- $R=0$ a $S=0$.
- **$R=1$ a $S=1$.**
- se R a S nerovnejí.
- je R nebo S nenastaveno.

Klopný obvod je název obvodu

- ze skupiny sčítaček.
- ze skupiny kombinačních logických obvodů.
- **ze skupiny sekvenčních logických obvodů.**

Sčítačka pro jeden řád BCD kódu se realizuje pomocí dvou čtyřbitových sčítaček. Pokud je součet dvou BCD číslic klasickou sčítačkou větší než 9

- **provádí se korekce přičtením čísla 6.**
- provádí se korekce extrakcí dolních 4 bitů.
- není třeba dělat korekci, přenos se použije jako číslice vyššího řádu.

Žádný bit se neztrácí při

- logickém posunu bitů.
- **rotaci bitů.**
- aritmetickém posunu doleva.

Násobení dvěma lze realizovat

- rotací o jeden bit doprava.
- aritmetickým posunem o jeden bit doprava.
- **aritmetickým posunem o jeden bit doleva.**

Operaci celočíselného dělení dvěma lze provést

- aritmetickým posuvem obsahu registru doleva
- logický posuvem obsahu registru doleva
- logický posuvem obsahu registru doprava
- **aritmetickým posuvem obsahu registru doprava**

Co není správně?

- Boolova algebra je nauka o operacích na dvouprvkové množině
- Boolova algebra užívá tři základní operace
- **Boolova algebra je vybudována na operaci negovaného logického součinu**

Technologie TTL používá jako svůj základní prvek

- **tranzistor NPN**
- tranzistor PNP
- invertor
- magnetické obvody

Pro technickou realizaci je nejméně vhodná

- **Booleova algebra**
- Pierceova algebra
- Shefferova algebra
- všechny algebry jsou stejně vhodné

Shefferova algebra je vybudována pouze na jediné logické operaci, a to

- **NAND**
- NOR
- XOR
- NOXOR
- AND

Piercova algebra je vybudována pouze na jediné logické operaci, a to

- NAND
- **NOR**
- XOR
- NOXOR
- OR

Základním stavebním prvkem technologie TTL je

- relé
- elektronka
- unipolární tranzistor
- **bipolární tranzistor**

Logický obvod NAND

- pro vstupy 0 a 0 dá výstup 0
- **pro vstupy 0 a 0 dá výstup 1**
- pro vstupy 0 a 1 dá výstup 0
- pro vstupy 1 a 1 dá výstup 1
- provádí negaci logického součtu

Logický obvod NOR

- pro vstupy 0 a 0 dá výstup 0
- pro vstupy 0 a 1 dá výstup 1
- **pro vstupy 1 a 0 dá výstup 0**
- pro vstupy 1 a 1 dá výstup 1
- provádí negaci logického součinu

Logický obvod XOR (nonekvivalence)

- **pro vstupy 0 a 0 dá na výstup 0**
- pro vstupy 0 a 1 dá na výstup 0
- pro vstupy 1 a 1 dá na výstup 1
- pro vstupy 0 a 0 dá na výstup 1
- provádí negaci vstupu

Negaci bitu provádí:

- logický obvod AND
- logický obvod OR
- **invertor**
- multiplexor
- dekodér

Pro výběr jednoho z n vstupů slouží:

- logický obvod AND
- logický obvod NOR
- invertor
- **multiplexor**
- dekodér

n adresových vstupů a 2^n datových výstupů má:

- logický obvod AND
- logický obvod NOR
- invertor
- multiplexor
- **dekodér**

Impuls je

- trvalá změna hodnoty signálu
- **dočasná změna hodnoty signálu**
- invertování hodnoty bitu

Mezi sekvenční logické obvody patří

- multiplexor, dekodér, sčítačka modulo 2
- polosčítačka, klopný obvod JK, klopný obvod RS
- **klopný obvod JK, klopný obvod RS, klopný obvod D**
- žádná z uvedených možností

Zakázaný stav se nachází u

- u polosčítačky
- klopného obvodu D
- klopného obvodu JK
- **žádná z uvedených možností**

Sekvenční logické obvody se vyznačují tím, že

- výstup nezávisí na předchozí posloupnosti změn
- nemají vnitřní paměť
- **výstup závisí na předchozí posloupnosti změn**
- nemají tzv. zpětnou vazbu

Výstupy z eventuální sčítačky Modulo 4 mohou nabývat hodnoty

- 0, 1
- 0, 1, 2
- **0, 1, 2, 3**
- 0, 1, 2, 3, 4

Pro kombinační logické obvody platí, že

- nepatří sem sčítačka modulo 2
- **výstupy nezávisí na předchozí posloupnosti změn**
- patří sem klopný obvod RS
- výstupy závisí na předchozí posloupnosti změn

Signálem Reset

- **je návrat do předem definovaného stavu**
- není návrat do předem definovaného stavu
- vynulujeme všechny výstupní hodnoty
- všem vstupním hodnotám přiřadíme jedničku

Mezi kombinační logické obvody nepatří

- polosčítačka
- multiplexor
- sčítačka modulo 2
- **žádná z uvedených možností**

Zakázaný stav klopného obvodu JK nastane když

- $J=0, K=0$
- $J=1, K=1$
- $J=1, K=0$
- **žádná z uvedených možností**

Korekce pro BCD sčítačku nepřičítá šestku, když

- bity součtu binárního řádu 1 a 3 jsou rovny jedné
- bity součtu binárního řádu 2 a 3 jsou rovny jedné
- přenosový bit součtu je roven jedné
- **přenosový bit součtu je roven nule**

Logický posun nenulového obsahu registru doprava

- nikdy neovlivní znaménko
- nejvyššímu bitu přiřadí jedničku
- **nejnižší bit se ztrácí**
- žádná z uvedených možností

Aritmetický posun nenulového obsahu registru doleva způsobí

- obsah registru se celočíselně vydělí dvěma, nezmění se znaménko, nedošlo-li k přetečení
- **obsah registru se celočíselně vynásobí dvěma, nezmění se znaménko, nedošlo-li k přetečení**
- obsah registru ani znaménko se nezmění
- obsah registru i znaménko se změní, pokud nedošlo k přetečení

Pokud se obsah registru posune aritmeticky doprava a číslo se blíží k maximální hodnotě, kterou lze do registru uložit, pak

- **obsah bude celočíselně vydělen dvěma**
- obsah bude vynásoben dvěma a výsledek bude správný
- obsah registru přeteče
- žádná z uvedených možností

Jednotka Baud udává

- počet bajtů přenesených za sekundu
- počet bitů přenesených za sekundu
- **počet změn stavů přenesených za sekundu**

Při stejné přenosové rychlosti je vždy počet bitů přenesených za sekundu

- menší nebo roven počtu baudů
- **větší nebo roven počtu baudů**
- menší než počet baudů
- větší než počet baudů
- rovný počtu baudů

Jako tzv. hradlo funguje

- **součinnový logický člen**
- součtový logický člen
- logický člen NOR
- logický člen nonekvivalence
- invertor

Jako sčítačka modulo 2, která neřeší přenosy, funguje

- logický člen NOR
- logický člen NAND
- **logický člen XOR**
- klopný obvod D
- klopný obvod RS

Polosčítačka se dvěma vstupy

- má tři výstupy
- řeší přenos z nižšího řádu
- její pravdivostní tabulka má 8 řádků
- **dává na výstup přenos do vyššího řádu**

Klopný obvod RS řízený nulami

- nemá zakázaný stav
- nemá definovaný stav pro vstupy 1 a 1
- **pro hodnoty 1 a 1 setrvává v předchozím stavu**
- pro hodnoty 0 a 0 setrvává v předchozím stavu

"R" v názvu klopného obvodu RS znamená

- repeat
- **reset**
- read
- random
- ready

Registry jsou typicky konstruovány z

- **klopného obvodu D**
- klopného obvodu JK
- klopného obvodu RS
- poloscítačky
- úplné scítačky

Při dvoustavové komunikaci je rychlost přenosu udávaná v baudech (Bd)

- větší než rychlost udávaná v bitech za sekundu
- menší než rychlost udávaná v bitech za sekundu
- **stejná jako rychlost udávaná v bitech za sekundu**
- neporovnatelná s rychlostí udávanou v bitech za sekundu

Při čtyřstavové komunikaci je rychlost přenosu udávaná v baudech (Bd)

- větší než rychlost udávaná v bitech za sekundu
- **menší než rychlost udávaná v bitech za sekundu**
- stejná jako rychlost udávaná v bitech za sekundu
- neporovnatelná s rychlostí udávanou v bitech za sekundu

Pod pojmem "zakázané pásmo" při přenosu signálu rozumíme

- skupinu počítačů, ke kterým signál nesmí dorazit
- frekvenci, se kterou nesmí vysílající vysílat
- **rozsah napětí, v jehož rámci je hodnota signálu nedefinovaná**
- všechny hodnoty napětí nerovnající se U_l a U_h

Pro multiplexor neplatí

- má datové vstupy
- má adresové vstupy
- má datový výstup
- **má adresový výstup**

Jaký zakázaný stav má klopný obvod RS řízený jedničkami?

- 0,0
- 0,1
- 1,0
- **1,1**

Pod rotací bitů vlevo rozumíme

- **posuv z nižšího řádu do vyššího, žádná hodnota bitu se neztrácí**
- posuv z nižšího řádu do vyššího, ztrácí se hodnota některého bitu
- posuv z vyššího řádu do nižšího, žádná hodnota bitu se neztrácí
- posuv z vyššího řádu do nižšího, ztrácí se hodnota některého bitu

Pod rotací bitů vpravo rozumíme

- posuv z nižšího řádu do vyššího, žádná hodnota bitu se neztrácí
- posuv z nižšího řádu do vyššího, ztrácí se hodnota některého bitu
- **posuv z vyššího řádu do nižšího, žádná hodnota bitu se neztrácí**
- posuv z vyššího řádu do nižšího, ztrácí se hodnota některého bitu

Pod pojmem logický posun vlevo rozumíme

- posuv z nižšího řádu do vyššího, žádná hodnota bitu se neztrácí
- **posuv z nižšího řádu do vyššího, ztrácí se hodnota některého bitu**
- posuv z vyššího řádu do nižšího, žádná hodnota bitu se neztrácí
- posuv z vyššího řádu do nižšího, ztrácí se hodnota některého bitu

Pod pojmem logický posun vpravo rozumíme

- posuv z nižšího řádu do vyššího, žádná hodnota bitu se neztrácí
- posuv z nižšího řádu do vyššího, ztrácí se hodnota některého bitu
- posuv z vyššího řádu do nižšího, žádná hodnota bitu se neztrácí
- **posuv z vyššího řádu do nižšího, ztrácí se hodnota některého bitu**

Při aritmetickém posunu

- se mění hodnota znaménkového bitu, nedojde-li k přetečení
- **se nemění hodnota znaménkového bitu, nedojde-li k přetečení**
- je posun doleva ekvivalentní celočíselnému dělení dvěma
- je posun doprava ekvivalentní násobení dvěma

V technologii TTL při použití tranzistoru NPN se kolektor a emitor otevírá

- **když je na bázi přivedena vysoká úroveň -- logická jednička**
- když je na bázi přivedena nízká úroveň -- logická nula
- když je na kolektor přivedena vysoká úroveň -- logická jednička
- když je na kolektor přivedena nízká úroveň -- logická nula

K čemu se využívá Karnaughova mapa

- **k minimalizaci počtu operací B-algebry**
- k uchování informace o rámcích, které nejsou zaplněny
- k uchování informace o dostupných V/V branách
- pro popis volných bloků paměti

Pokud jsou 1 a 1 na vstupu sčítačky modulo 2, pak na výstupu je

- **0**
- 1
- 2
- tento vstup je neplatný

Mám 16 zařízení, zařízení číslo 10 chci poslat signál 1, ostatním 0. Co použiji?

- **dekodér**
- multiplexor
- úplnou sčítačku
- poloSčítačku

Pro úplnou sčítačku pro jeden binární řád platí

- **má 3 vstupy a 2 výstupy**
- má 2 vstupy a 3 výstupy
- má 2 vstupy a 2 výstupy
- má 3 vstupy a 3 výstupy

Co platí pro klopný obvod D?

- **je to paměť na jeden bit**
- má čtyři výstupy
- má čtyři datové vstupy
- má ekvivalentní funkci jako poloSčítačka

NOXOR je stejný jako:

- **ekvivalence**
- NOR
- OR
- NAND

Které zapojení nelze popsat pomocí Booleovy algebry?

- sériové
- **místkové**
- paralelní
- sérioparalelní

Která paměť musí být energeticky nezávislá?

- **vnější paměť**
- vnitřní paměť
- registry

Obsah adresového registru paměti se na výběr jednoho z výběrových (adresových) vodičů převádí

- multiplexorem 1 z N.
- **dekodérem 1 z N.**
- sčítačkou 1 plus N.

K destruktivnímu nevratnému zápisu do permanentní paměti pomocí přepalování tavných spojek proudovými impulsy je určena paměť

- ROM
- **PROM**
- EPROM

Parametr pamětí "vybavovací doba - čas přístupu" bude nejvyšší u

- registru
- vyrovnávací (cache) paměti
- operační paměti
- **diskové paměti**

Paměť, která svůj obsah adresuje klíčem, který je uložen odděleně od obsahu paměti a vyhledává se v klíči paralelně, se nazývá

- operační paměť.
- permanentní paměť.
- **asociativní paměť.**
- klíčová paměť.

Paměť typu cache nebývá umístěna mezi

- procesorem a pamětí
- procesorem a V/V zařízením
- **procesorem a registry**

Do paměti typu PROM

- nelze data zapsat
- **lze zapsat data pouze jednou**
- lze zapsat data libovolněkrát působením UV záření
- lze zapsat data libovolněkrát vyšší hodnotou elektrického proudu
- lze zapsat data libovolněkrát přepálením tavné pojistky NiCr

Které tvrzení neplatí pro popis fyzické struktury vnitřní paměti?

- Dekodér na jeden z adresových vodičů nastaví hodnotu logická 1.
- Informace je na koncích datových vodičů zesílena zesilovačem.
- Adresa je přivedena na vstup dekodéru.
- Podle zapojení buněk na řádku projde/neprojde logická 1 na datové vodiče.
- **Datový registr má na vstup přivedeny adresové vodiče.**

Máme-li vnitřní paměť o kapacitě 16 bitů zapojenou jako matici paměťových buněk 4x4 bity, pak nejmenší adresovatelná jednotka je

- 1 bit
- 2 bity
- **4 bity**
- 16 bitů
- 65536 bitů

Působením UV záření je možné vymazat obsah paměti

- ROM
- PROM
- **EPROM**
- EEPROM
- RAM

Statickou, energeticky nezávislou pamětí není paměť typu

- ROM
- PROM
- EPROM
- EEPROM
- **žádná z odpovědí není správně**

Vybavovací doba paměti znamená

- **čas přístupu k jednomu záznamu v paměti**
- doba potřebná pro přenesení 1 KB dat do paměti
- čas potřebný pro instalaci paměťového modulu
- doba potřebná pro načtení celé kapacity paměti

Pro paměť s přímým přístupem platí

- musím se k informaci "pročíst", doba přístupu není konstantní
- **doba přístupu k libovolnému místu v paměti je konstantní**
- obsah z adres nižších hodnot získám rychleji než vyšších

Energeticky závislá paměť obecně obsahuje po obnově napájení

- předdefinovaný konstantní obsah
- samé nuly
- samé jedničky
- **obsah paměti je nedefinovaný**

Energeticky závislá paměť typicky je

- **paměť RAM**
- harddisk
- paměť Flash
- CD-R

Správný postup čtení dat z paměti je

- procesor vloží adresu do adresového registru, příkaz čti, procesor převezme informaci z datového registru
- procesor vloží adresu do datového registru, příkaz čti, procesor převezme informaci z datového registru
- procesor vloží adresu do adresového registru, procesor zapíše informaci z datového registru, příkaz čti
- žádná z uvedených možností neplatí

Paměť určená pro čtení i pro zápis má zkratku

- ROM
- PROM
- EPROM
- **RWM**

Zpětnému proudu v ROM pamětech zabráňuje

- použití vodičů
- **použití polovodičů**
- použití nevodičů
- žádná z uvedených možností

Kolikrát je možno zapisovat do paměti PROM?

- pouze při výrobě
- **lze jednou naprogramovat**
- lze přeprogramovat libovolněkrát

Ultrafialovým světlem lze přemazat paměť

- ROM
- PROM
- **EPROM**
- RWM

Elektrickým proudem lze přemazat paměť

- ROM
- PROM
- EPROM
- **EEPROM**

Paměť, ze které se většinou čte, maže se elektrickým proudem a dá se do ní i zapisovat má zkratku

- **RMM**
- RWM
- ROM
- RUM

Pro asociativní paměť neplatí

- v paměti se plní klíč a obsah
- paměť klíčů se prohledává paralelně
- zkratka je CAM
- **používá se jako operační paměť**

CAM paměti předám adresu. Nejdříve ji hledá v

- adresovém registru
- datovém registru
- obsahu ke klíčům
- **paměti klíčů**

Jaké sběrnice jsou mezi procesorem a pamětí?

- pouze datová
- pouze adresová
- **datová a adresová**
- datová, adresová a pro v/v zařízení

Jakou funkci u paměti má refresh cyklus?

- jednorázově vymaže obsah paměti
- **obnovuje data uložená v dynamické paměti**
- obnovuje data uložená ve statické paměti
- opraví chybu v paměti

Mezi paměti s výhradně s přímým přístupem patří

- páska
- disk
- **operační paměť**

Která z uvedených pamětí není programovatelná?

- **ROM**
- PROM
- EPROM
- EEPROM

Pro statickou paměť neplatí

- informace se udržuje, pokud je napájení
- **informace se udržuje, i když není napájení**
- informace se neudržuje, když není napájení

ROM je paměť

- pouze pro zápis
- **pouze pro čtení**
- pro zápis i pro čtení
- žádná odpověď není správná

ROM je zkratka pro

- **read only memory**
- read on memory
- read only matter
- ride on memory

Páska je paměť

- **se sekvenčním přístupem**
- s přímým přístupem
- s kombinovaným přístupem
- s indexsekvenčním přístupem

Na libovolnou adresu v paměti s přímým přístupem se dostanu typicky

- za proměnlivý čas
- **za konstantní čas**
- záleží na nastavení v operačním systému
- nelze jednoznačně určit

PC -> AR, 0 -> WR, DR -> IR

PC+1 -> AR, 0 -> WR, DR -> TA_L

PC+2 -> AR, 0 -> WR, DR -> TA_H

TA -> AR, 0 -> WR, DR -> A

PC+2 -> PC

- jsou mikroinstrukce instrukce LDA
- jsou mikroinstrukce instrukce STA
- jsou mikroinstrukce jiné instrukce
- **tyto mikroinstrukce jsou nekorektní**

Mezi aritmetické instrukce fiktivního procesoru definovaného na přednáškách patří pouze tyto

- ADD, MOV, CMP
- STA, ADD, CMA
- **ADD, CMA, INR**

Příznaky pro větvení programu vždy nastavují tyto instrukce fiktivního procesoru definovaného na přednáškách

- **ADD, INR, CMA**
- LDA, ADD, CMP
- ADD, MOV, INR

Příznaky pro větvení programu nikdy nemění tyto instrukce fiktivního procesoru definovaného na přednáškách

- CMA, JMP, LDA
- **MOV, STA, JMP**
- STA, LDA, CMP

Registr PC -- čítač instrukcí v procesoru obsahuje

- **adresu právě prováděné instrukce.**
- počet již provedených instrukcí.
- počet instrukcí, které zbývají do konce programu.

Jednou z fází zpracování instrukce procesorem není:

- výběr operačního kódu z paměti
- výběr adresy operandu z paměti
- **kopírování instrukce do paměti**
- provedení instrukce
- zápis výsledků zpracované instrukce

Pro adresaci operační paměti mající kapacitu 64 K adresovatelných jednotek (bajtů) je třeba adresová sběrnice šířky

- 10 bitů.
- **16 bitů.**
- 20 bitů.
- 32 bitů.

Pro adresaci operační paměti mající kapacitu 1 M adresovatelných jednotek (bajtů) je třeba adresová sběrnice šířky

- 10 bitů.
- 16 bitů.
- **20 bitů.**
- 32 bitů.

Instrukce mající zkratku LDA typicky znamená

- ulož obsah registru A do paměti na adresu zadanou operandem instrukce.
- vynuluj obsah registru A.
- zvyš obsah registru A o jedničku.
- **naplň obsah registru A hodnotou z paměti.**

Instrukce mající zkratku JMP typicky provádí

- **nepodmíněný skok.**
- podmíněný skok na adresu zadanou operandem.
- volání podprogramu.

Příznakový registr procesoru se používá na

- sledování výkonnosti procesoru.
- **realizaci podmíněných skoků.**
- zaznamenávání verzí firmware procesoru.

Instrukce CMP pro porovnání typicky

- větší číslo uloží do registru A.
- uloží do registru A hodnotu 1, pokud je první číslo větší.
- **pouze nastaví příznaky.**

Posloupnost instrukcí

```
LDA x
MOV B,A
LDA y
CMP B
JP ne
ano: ...
JMP ven
ne: ...
ven: ...
```

vyjadřuje příkaz

- **IF x>y THEN ano ELSE ne;**
- IF x>=y THEN ano ELSE ne;
- IF x<y THEN ano ELSE ne;
- IF x<=y THEN ano ELSE ne;

Posloupnost instrukcí

```
LDA y
MOV B,A
LDA x
CMP B
JM ne
ano: ...
JMP ven
ne: ...
ven: ...
```

vyjadřuje příkaz

- IF x>y THEN ano ELSE ne;
- **IF x>=y THEN ano ELSE ne;**
- IF x<y THEN ano ELSE ne;
- IF x<=y THEN ano ELSE ne;

Posloupnost instrukcí

```
LDA y
MOV B,A
LDA x
CMP B
JP ne
ano: ...
JMP ven
ne: ...
ven: ...
```

vyjadřuje příkaz

- IF x>y THEN ano ELSE ne;
- IF x>=y THEN ano ELSE ne;
- **IF x<y THEN ano ELSE ne;**
- IF x<=y THEN ano ELSE ne;

Posloupnost instrukcí

```
LDA x
MOV B,A
LDA y
CMP B
JM ne
ano: ...
JMP ven
ne: ...
ven: ...
```

vyjadřuje příkaz

- IF x>y THEN ano ELSE ne;
- IF x>=y THEN ano ELSE ne;
- IF x<y THEN ano ELSE ne;
- **IF x<=y THEN ano ELSE ne;**

```
PC -> AR, 0 -> WR, DR -> IR
PC+1 -> AR, 0 -> WR, DR -> TAL
PC+2 -> AR, 0 -> WR, DR -> TAH
TA -> AR, 0 -> WR, DR -> TAXL
TA+1 -> AR, 0 -> WR, DR -> TAXH
TAX -> AR, A -> DR, 1 -> WR
PC+3 -> PC
```

- jsou mikroinstrukce instrukce LDA
- jsou mikroinstrukce instrukce STA
- jsou mikroinstrukce LDAX (nepřímé naplnění)
- **jsou mikroinstrukce STAX (nepřímé naplnění)**
- tyto mikroinstrukce jsou nekorektní

Instrukce podmíněného skoku

- provede následující instrukci, pokud je splněna podmínka.
- skočí na instrukci, jejíž adresa je zadána operandem, pokud podmínka není splněna.
- **provede následující instrukci, pokud podmínka splněna není.**

Operace PUSH nad zásobníkem

- **vloží položku do zásobníku.**
- vybere položku ze zásobníku.
- stlačí obsah zásobníku.

Jaký je správný postup operací?

- **PUSH sníží SP a uloží položku na adresu podle SP; POP vybere z adresy podle SP a zvýší SP.**
- PUSH sníží SP a uloží položku na adresu podle SP; POP zvýší SP a vybere z adresy podle SP.
- PUSH uloží položku na adresu podle SP a sníží SP; POP vybere z adresy podle SP a zvýší SP.

Instrukce volání podprogramu musí

- **uchovat návratovou adresu.**
- uchovat obsah čítače instrukcí.
- uchovat obsah registrů do zásobníku.

Pojem 'time-out' při provádění V/V operací znamená, že např.

- **zahájená výstupní operace neodpověděla 'hotovo' do definované doby.**
- mezi výstupní a vstupní operací musí být prodleva definované doby.
- před zahájením vstupní operace lze signál 'start' poslat ne dříve než uplyne definovaná doba.

Posloupnost instrukcí

START

opak: FLAG opak

IN

STA x

je podle toho, jak jsme si na přednáškách definovali vlastní procesor (pomíjíme otázku time-outu, neefektivního využití procesoru),

- **korektní operace čtení ze vstupního zařízení**
- korektní operace zápisu do výstupního zařízení
- žádná z ostatních odpovědí není správná

Posloupnost instrukcí

LDA x

START

OUT

opak: FLAG opak

je podle toho, jak jsme si na přednáškách definovali vlastní procesor (pomíjíme otázku time-outu, neefektivního využití procesoru),

- korektní operace čtení ze vstupního zařízení
- korektní operace zápisu do výstupního zařízení
- **žádná z ostatních odpovědí není správná**

Ve kterém z následujících okamžiků by mělo dojít ke vzniku přerušení?

- zahájení tisku znaku
- **konec tisku znaku**
- ukončení programu

Které z konstatování vztahujících se k okamžiku přerušení procesu je nesprávné?

- Přerušit nelze během provádění instrukce.
- Přerušit lze pouze tehdy, je-li to povoleno (nejde-li o nemaskovatelné přerušení).
- Přerušit nelze bezprostředně po zahájení obsluhy přerušení.
- **Přerušení nastane ihned po žádosti signálem INTERRUPT.**

Jaké je správné modelové chování obsluhy vzniku přerušení?

- **Mikroinstrukce musí uložit PC a vynulovat IF. Programem se ukládají všeobecné registry.**
- Mikroinstrukce musí uložit PC a všeobecné registry. Program dle svého zvážení vynuluje IF.
- Mikroinstrukce uloží obsah PC. Program uloží dle zvážení obsah všeobecných registrů a vynuluje IF.

Operační kód (operační znak) je

- numerické vyjádření konkrétní instrukce, je vždy stejně dlouhý
- **numerické vyjádření konkrétní instrukce, má typicky proměnlivou délku**
- je adresa operandu
- je adresa 1. a 2. operandu

Operační kód není

- operační znak
- numerické vyjádření konkrétní instrukce, které má proměnlivou délku
- součást instrukce
- **žádná z uvedených možností**

Pro čítač instrukcí procesoru neplatí

- může mít zkratku PC
- může mít zkratku IP
- obsahuje adresu prováděné instrukce
- **žádná z uvedených možností**

Která instrukce naplní registr A obsahem slabiky z paměti?

- STA
- **LDA**
- INA
- JMP

Instrukce STA

- **uloží registr A do paměti**
- naplní registr A obsahem slabiky z paměti
- je nepodmíněný skok na adresu A
- žádná z uvedených možností

Instrukce JMP je

- **nepodmíněný skok**
- podmíněný skok
- uloží registr P do paměti
- žádná z uvedených možností

Osmibitový procesor se 64KB pamětí má

- 8bitovou datovou sběrnici a 20bitovou adresovou sběrnici
- 8bitovou datovou sběrnici a 8bitovou adresovou sběrnici
- **8bitovou datovou sběrnici a 16bitovou adresovou sběrnici**

Registr PC procesoru naplníme instrukcí

- LDA
- STA
- **JMP**
- žádnou z uvedených

Pomocný 16bitový registr TA procesoru definovaného na přednáškách se skládá z

- **8bitového TA High a 8bitového TA Low**
- 12bitového TA High a 4bitového TA Low
- 4bitového TA High a 12bitového TA Low
- žádná z uvedených možností

První fázi každé instrukce je

- výběr operandu
- provedení instrukce
- **výběr operačního znaku**
- aktualizace PC

Pro mikroinstrukci výběr operačního znaku neplatí

- cílem je vložit do instrukčního registru instrukci
- je vždy 1. fází instrukce
- **cílem je vložit do datového registru data**
- je součástí např. instrukce LDA

Mikroinstrukce výběr operačního znaku znamená

- **procesor zjistí, kterou instrukci provádí**
- procesor načte adresu z adresového registru
- procesor zahájí instrukci LDA
- žádná z uvedených možností

Mezi mikroinstrukce instrukce LDA nepatří

- výběr operačního znaku
- výběr operandu
- aktualizace registru PC zvýšením o délku instrukce
- **naplnění registru PC hodnotou operandu instrukce**

Instrukce INR procesoru definovaného na přednáškách způsobí

- **zvýší obsah registru o jedna**
- sníží obsah registru o jedna
- uloží obsah registru R do paměti
- načte obsah registru R z paměti

Instrukce CMA procesoru definovaného na přednáškách způsobí

- **inverzi bitů v registru A**
- zvýší obsah registru A o jedna
- sníží obsah jedničku A o jedna
- žádná z uvedených možností

Která instrukce sníží obsah registru o jedna

- INR
- CMA
- ADD
- **žádná z uvedených možností**

Instrukce ADD procesoru definovaného na přednáškách

- **přičte obsah registru k registru A**
- invertuje bity v registru A
- vždy zvýší obsah registru A o jedna
- žádná z uvedených možností

Příznak procesoru definovaného na přednáškách není

- jednobitový indikátor
- Z (zero)
- CY (Carry)
- **žádná z uvedených možností**

S (Sign) je příznak procesoru definovaného na přednáškách, kterým je

- **kopie znaménkového bitu výsledku operace**
- kopie znaménkového bitu 1. operandu
- kopie znaménkového bitu 2. operandu
- 1 při nulovém výsledku operace

Pro příznaky procesoru definovaného na přednáškách platí

- nastavuje je programátor
- **nastavuje je procesor**
- nastavuje je procesor a programátor může nastavování vypnout
- žádná z uvedených možností

Příznaky procesoru definovaného na přednáškách mění instrukce

- **INR, ADD, CMA**
- LDA, STA
- LDA, STA, JMP
- LDA, STA, JMP, MOV

Instrukce procesoru definovaného na přednáškách CMP B porovná obsah registru A s obsahem registru B a

- **změní podle toho příznaky**
- nezmění podle toho příznaky
- uloží výsledek do registru A
- uloží výsledek do registru B

Mezi příznaky procesoru definovaného na přednáškách nepatří

- CY
- AC
- **TA**
- Z

Změnu znaménka u čísla v registru A procesoru definovaného na přednáškách provedeme posloupností instrukcí

- **CMA, INR A**
- CMA, MOV B,A
- INR A, CMA
- žádná z uvedených možností

Pro zásobník procesoru definovaného na přednáškách neplatí, že

- je datová struktura fungující systémem LIFO
- **je datová struktura fungující systémem FIFO**
- vkládá se do ní operací PUSH
- vybírá se z ní operací POP

PUSH procesoru definovaného na přednáškách

- **je instrukce, vkládá obsah registru do zásobníku**
- je instrukce, vybírá obsah ze zásobníku
- je příznak
- je interní registr

PSW procesoru definovaného na přednáškách je

- **stavové slovo procesoru, tvořeno z registru A a příznaků**
- stavové slovo procesoru, tvořeno z registru A
- stavové slovo procesoru, tvořeno z příznaku na předdefinovaný registr
- žádná z uvedených možností

Pro zásobník procesoru definovaného na přednáškách platí

- má kontrolu podtečení
- **nemá kontrolu podtečení**
- je strukturou First in First out
- žádná z uvedených možností

LXISP procesoru definovaného na přednáškách

- je ukazatel na vrchol zásobníku
- zapisuje hodnotu na dno zásobníku
- **definuje dno zásobníku**
- instrukce, která vkládá obsah registru A do zásobníku

Instrukce PUSH procesoru definovaného na přednáškách

- **numericky snižuje ukazatel vrcholu zásobníku**
- numericky zvyšuje ukazatel vrcholu zásobníku
- inkrementuje SP
- žádná z uvedených možností

Instrukce POP procesoru definovaného na přednáškách

- definuje dno zásobníku
- snižuje ukazatel vrcholu zásobníku
- dekrementuje SP
- **žádná z uvedených možností**

Pro instrukci RET procesoru definovaného na přednáškách neplatí

- vrátí se z podprogramu do těla programu
- obsah vrcholu zásobníku je vložen do registru PC
- **vrátí se na absolutní začátek programu**
- používá se na konci podprogramu

Která posloupnost instrukcí může korektně obsloužit time-out při programování V/V operace procesoru definovaného na přednáškách

- **100 START**
- 100 START
- 100 START

Instrukce OUT procesoru definovaného na přednáškách

- **zapiše obsah reg. A na datovou sběrnici pro v/v zařízení**
- načte obsah datové sběrnice od v/v zařízení a uloží jej do A
- zapisuje obsah reg. A a zahájí vstupně výstupní operaci

Která instrukce procesoru definovaného na přednáškách skočí na adresu, není-li operace hotova?

- START
- **FLAG**
- IN
- OUT

Posloupnost instrukcí procesoru definovaného na přednáškách LDA x, OUT, START, FLAG je

- korektní operace čtení ze vstupního zařízení
- **korektní operace zápisu do výstupního zařízení**
- žádná z ostatních odpovědí není správná

Posloupnost instrukcí procesoru definovaného na přednáškách START, IN, STA x, FLAG je

- korektní operace čtení ze vstupního zařízení
- korektní operace zápisu do výstupního zařízení
- **žádná z ostatních odpovědí není správná**

Co je time-out?

- **doba, kterou jsme ochotni čekat na dokončení V/V operace**
- doba, kterou jsme ochotni čekat na začátek V/V operace
- doba, kterou nemůžeme ovlivnit (je předdefinovaná)

Signál INTERRUPT (INTR)

- **žádá o přerušení v procesoru**
- deaktivuje rutinu pro obsluhu přerušení
- žádá o ukončení provádění procesu
- žádá o uvedení procesoru do počátečních podmínek

Která činnost se vykonává jako poslední při návratu z přerušení procesoru definovaného na přednáškách?

- provedení obslužné rutiny, která zjistí kdo žádá o přerušení
- přerušení provádění programu
- **obnovení PC, A, ...**
- úklid obsahu registrů PC, A, ...

Pro přerušení platí:

- přerušit lze pouze během provádění instrukce
- lze přerušit bezprostředně po zahájení obsluhy předchozího přerušení
- **o přerušení se musí požádat signálem INTERRUPT**
- přerušení se používá typicky v kritické sekci

Instrukce, která zakáže přerušení procesoru definovaného na přednáškách, se nazývá

- STI
- **CLI**
- INTERRUPT
- žádná možnost není správná

Co je v registru PC procesoru definovaného na přednáškách při uplatnění žádosti o přerušení

- adresa instrukce, která byla provedena před přerušením
- **adresa instrukce, která nebyla provedena v důsledku přerušení**
- adresa vrcholu zásobníku

Během uplatnění přerušení není provedeno

- uložení registru PC do zásobníku
- vynulování IF
- **povolení přerušení**
- uklizení registru A a dalších do zásobníku

Která z instrukcí nepatří mezi instrukce procesoru definovaného na přednáškách, které se použijí při návratu z přerušení

- POP
- STI
- RET
- **CLI**

Co neplatí pro instrukci STI procesoru definovaného na přednáškách

- povolí přerušení až po provedení následující instrukce
- nastaví IF na hodnotu 1
- **povolí přerušení po svém dokončení**

Signál RESET procesoru definovaného na přednáškách nezpůsobí

- nastavení procesoru do počátečních podmínek
- **vynulování příznaků procesoru**
- předání řízení na adresu ukazující zpravidla do permanentní paměti
- zakázání přerušení
- vynulování IF

Pro signál RESET procesoru definovaného na přednáškách neplatí

- provede se kdykoliv
- nastaví IF na nulu
- **provede se pouze při přerušení**
- předá řízení na adresu ukazující zpravidla do v permanentní paměti

Výběr instrukcí procesoru definovaného na přednáškách je řízen registrem

- **PC**
- AR
- DR
- IR

Který z registrů procesoru definovaného na přednáškách není 16bitový

- PC
- **IR**
- TA
- AR

Která instrukce procesoru definovaného na přednáškách nenastavuje příznaky

- INR
- ADD
- **LDA**
- CMA

Která instrukce procesoru definovaného na přednáškách nastavuje příznaky

- LDA
- **ADD**
- STA
- JMP

Která instrukce procesoru definovaného Pokud používáme virtualizaci paměti, na přednáškách porovná zadaný registr pak s registrem A

- CMA
- **CMP**
- STA
- LDA

Zásobník má strukturu

- **LIFO**
- FIFO
- PIFO
- SIFO

Fronta má strukturu

- LIFO
- **FIFO**
- PIFO
- SIFO

Pro instrukci CALL procesoru definovaného na přednáškách neplatí

- uloží návratovou adresu do zásobníku
- provede nepodmíněný skok na zadanou adresu
- **přečte obsah zadaného registru**
- provede totéž co posloupnost instrukcí PUSH a JMP

Procesor rozlišuje komunikaci s pamětí a se V/V zařízeními

- užíváním různých sběrnic
- **signálem M/IO**
- signálem NMI
- signálem CLK

Jak široká musí být adresa, pokud chceme adresovat 1 K stránek a každá stránka má velikost 4 K adresovatelných jednotek.

- 12 bitů.
- 16 bitů.
- **22 bitů.**
- 32 bitů.

- **šířka virtuální adresy by měla být větší nebo rovna šířce reálné adresy.**
- šířka virtuální adresy by měla být menší nebo rovna šířce reálné adresy.
- se musí šířka virtuální adresy a reálné adresy shodovat.

K obecnému mechanismu virtuální paměti: Co je obvyklé?

- **Počet stránek je větší než počet rámců.**
- Počet stránek je roven počtu rámců.
- Počet stránek je menší než počet rámců.

K obecnému mechanismu virtuální paměti: Která z adres může být širší (má se na mysli, že je více bitová)

- reálná
- **virtuální**
- bezpodmínečně musí být reálná a virtuální adresa stejně velké

K obecnému mechanismu virtuální paměti: Co platí?

- Rámce jsou uloženy na disku, stránky jsou v reálné paměti.
- **Stránky jsou uloženy na disku, rámce jsou v reálné paměti.**

K obecnému mechanismu algoritmu LRU: Algoritmus LRU vybírá

- **nejdéle nepoužitou položku**
- nejméněkrát použitou položku
- nejdéle uloženou položku

Algoritmus LRU pro výběr oběti např. při virtualizaci paměti vybírá

- nejméněkrát použitý obsah rámce.
- **nejdéle nepoužitý obsah rámce.**
- náhodný rámec.
- předchozí použitý rámec.

Při virtualizaci paměti se používají pojmy

- segment a stránka.
- **rámec a stránka.**
- segment a rámec.

K obecnému mechanismu algoritmu LRU: K úplnému ošetření osmi položek algoritmem LRU (pomocí neúplné matice) bychom potřebovali kolik bitů v neúplné matici?

- **28**
- 36
- 24
- 16
- 8

Pro virtualizaci paměti neplatí

- paměť dělíme do rámců a disk na stránky
- reálná adresa ukazuje do reálné paměti
- počet stránek je větší nebo roven počtu rámců
- **rámec není stejně velký prostor jako stránka**

Při virtualizaci paměti neplatí

- obsah špinavého rámce musím před jeho smazáním zapsat na disk
- označení čistý rámec odpovídá označení rámec, do kterého nebylo zapsáno
- **rámec je špinavý, pokud má příznak parity nastaven na jedničku**
- do špinavého rámce bylo něco zapsáno

Co platí pro segmenty a stránky:

- **segmenty jsou různé velikosti, stránky jsou stejné velikosti**
- segmenty jsou stejné velikosti, stránky jsou různé velikosti
- segmenty jsou různé velikosti, stránky jsou různé velikosti
- segmenty jsou stejné velikosti, stránky jsou stejné velikosti

Co znamená LRU:

- **least recently used**
- last record used
- load record unsaved
- let ring upset

Jaká je nesprávná konfigurace virtuální paměti u obecného procesoru?

- 32bitová reálná adresa a 48bitová virtuální adresa
- **32bitová reálná adresa a 24bitová virtuální adresa**
- 24bitová reálná adresa a 24bitová virtuální adresa

Jaká je maximální hodnota adresy reálné paměti v procesoru Intel 8086

- **1048575₁₀**
- 1048576₁₀
- FFFF₁₆
- 10FFEF₁₆

Adresa 02AB:00A4₁₆ reálného režimu procesorů Intel se vyčíslí na hodnotu

- **2B54₁₆**
- 2CB4₁₆
- 34F₁₆
- 0CEB₁₆
- na žádnou z uvedených

Na jaké hodnoty se nastaví bity příznakového registru provedením instrukce ADD v procesorech Intel řady 86 s operandy -5 a 8

- **CF=1, ZF=SF=OF=0**
- CF=ZF=SF=OF=0
- CF=SF=1, ZF=OF=0
- ZF=CF=OF=1, SF=0

Jaký je korektní postup činností při přerušení v procesoru Intel 8086?

- **do zásobníku se uloží obsah reg. příznaků**
- vynulují se příznaky IF a TF
- ®ádný z uvedených.

Jaká je poslední (20bitová) adresa tabulky přerušovacích vektorů v procesoru Intel 8086 a reálných režimech procesorů vyšších

- **1023₁₀**
- 255₁₀
- 4095₁₀
- 0FFFFFFh

Adresový prostor adres V/V zařízení v procesorech Intel (typicky 8086) je

- 20bitový
- **16bitový**
- 8bitový

Kolik různých přerušení může vzniknout v procesoru Intel 8086 a reálných režimech procesorů vyšších

- **256**
- 128
- 1024
- 65536

Která z uvedených variant instrukce MOV v procesorech Intel je nekorektní?

- MOV prom1,AX
- **MOV prom1,prom2**
- MOV BX,prom2
- MOV AX,DX

Která z uvedených variant instrukce MOV v procesorech Intel je nekorektní?

- **MOV AL,BX**
- MOV CX,DX
- MOV CL,DH
- MOV BL,BL

Jaké dvě různé operace se v procesorech Intel realizují jedinou instrukcí?

- **SAL a SHL provádí SHL (arit. a logický posun bitů vlevo se vždy provádí jako logický posun vlevo)**
- SAL a SHL provádí SAL (arit. a logický posun bitů vlevo se vždy provádí jako aritm. posun vlevo)
- SAR a SHR provádí SAR (arit. a logický posun bitů vpravo se vždy provádí jako aritm. posun vpravo)
- SAR a SHR provádí SHR (arit. a logický posun bitů vpravo se vždy provádí jako logický posun vpravo)

Které varianty instrukce JMP v procesorech Intel přiřazují (nepřičítají) operand do registru IP?

- **vzdálený (far) skok a nepřímý skok**
- blízký (near) skok a nepřímý skok
- krátký (short) skok a blízký (near) skok
- vzdálený (far) skok a blízký (near) skok

Které varianty instrukce JMP v procesorech Intel přičítají (nepřiřazují) operand k obsahu registru IP?

- vzdálený (far) skok a nepřímý skok
- blízký (near) skok a nepřímý skok
- **krátký (short) skok a blízký (near) skok**
- vzdálený (far) skok a blízký (near) skok

Programujeme cyklus typu REPEAT, ve kterém na konci bloku testujeme, zda je hodnota i>5. Pokud ano, pak provádění bloku opakujeme. Neznáme však velikost bloku, který musíme opakovat. Blok začíná návěštím "Blok" a programujeme jej na procesoru Intel 8086. Jaká bude správná a nejbezpečnější realizace podmínky?

- CMP i,5
JG Blok

- **CMP i,5
JLE Dále
JMP Blok
Dále:**

- CMP i,5
JG Dále
JMP Blok
Dále:

Čím se procesor 8088 liší od procesoru 8086

- **8088 je určen pro vnější osmibitové prostředí**
- 8088 je 8bitový
- 8088 je 16bitový
- 8088 je 20bitový

NMI - nemaskovatelné přerušení se používá například při

- **hlášení chyb parity paměti**
- skocích z cyklu
- přechodu do reálného režimu
- žádostech o přerušení z rychlého zařízení (např. disku)

Při adresaci paměti procesoru 8086 neplatí

- používá se 20bitová adresa složená z dvou 16bitových komponent
- adresu zapisujeme ve tvaru segment: offset
- **používá se 32bitová adresa složená z dvou 16bitových komponent**

Mezi segmentové registry nepatří:

- CS
- **PC**
- SS
- DS

Pro registr CS platí

- **je určen pro výpočet adresy instrukce**
- slouží pro výpočet adresy dat
- je řídicím registrem
- je ekvivalentem registru PC

Pro registr IP neplatí

- je ekvivalentem registru PC
- obsahuje část adresy právě prováděné instrukce
- **obsahuje pomocný datový segment**

Adresu paměti u procesoru 8086 zapisujeme ve tvaru

- segment
- offset
- **segment: offset**
- offset: segment

Jakou velikost má jeden segment v procesoru 8086

- 16 bitů
- 20 KB
- **64 KB**
- 1 MB

Segment procesoru 8086 začíná na adrese dělitelné

- není specifikováno, smí začínat na libovolné
- **16**
- 4 K
- 32

Jaká je korektní posloupnost operací při uplatnění přerušení v procesoru 8086?

- IF:=0; PUSH F; PUSH CS; PUSH IP
- **PUSH F; IF:=0; PUSH CS; PUSH IP**
- PUSH AX; IF:=0; PUSH F; PUSH IP
- PUSH IP; PUSH AX; PUSH F; IF:=0

Instrukce IRET procesoru 8086 obnovuje ze zásobníku obsahy registrů

- IP, AX
- IP, CS
- **IP, CS, F**
- AX, CS, IP, F

Jaký rozsah adres v procesoru 8086 bude přepsán, pokud se v nekonečné smyčce zacyklí použití instrukce PUSH AX?

- 00000-FFFFF
- **SS:0000-SS:FFFF**
- CS:0000-CS:FFFF
- DS:0000-DS:FFFF

V trasovacím režimu (TF=1) procesoru 8086 se provedení jedné instrukce spustí instrukcí

- **IRET**
- JMP
- CALL
- RET

Trasovací režim procesoru 8086 se spouští nastavením TF=1

- instrukcí SETTF
- instrukcí CLTF
- **v příznaku TF**
- v registru TF

Trasovací režim procesoru 8086 se spouští nastavením TF=1 a ukončuje se

- instrukcí CLTF
- instrukcí STOPT
- **neukončuje se**

Důvodem, proč po použití instrukce MOV SS,... v procesoru 8086 se zakazuje přerušení na dobu provádění jedné následující instrukce, je

- časová náročnost instrukce MOV SS,...
- kontrola přetečení obsahu zásobníku
- **atomické naplnění adresy vrcholu zásobníku**
- odstranění zbývajících návratových adresy ze zásobníku

Nepovolená instrukce v procesoru 8086 je

- **MOV CS,...**
- MOV SS,...
- MOV DS,...
- MOV ES,...

Programátor procesoru 8086 nastavuje příznaky

- **DF, IF, TF**
- OF, SF, ZF
- AF, PF, CF

Příznak ZF procesoru 8086 je nastaven

- **při nulovém výsledku operace**
- při krokovacím režimu
- při aritmetickém přeplnění
- při sudé paritě výsledků

Příznak TF procesoru 8086

- **uvede procesor do krokovacího režimu**
- zabrání uplatnění vnějších maskovatelných přerušení
- je nastaven při nulovém výsledku operace

Všechny odkazy na zásobník procesoru 8086 jsou segmentovány přes registr

- **SS (Stack segment)**
- CS (Code segment)
- DS (Data segment)
- ES (Extra segment)

Pro vnější přerušení procesoru 8086 neplatí

- vyvolá se pomocí signálu INTERRUPT
- **vyvolá se např. při dělení nulou**
- vyvolá se pomocí signálu NMI
- dělí se na maskovatelná a nemaskovatelná

Pro vnitřní přerušení procesoru 8086 neplatí

- vyvolá se chybou při běhu programu
- je generováno programově
- vyvolá se instrukcí INT n
- **je generováno řadičem přerušení**

Akce, která se neprovádí při přerušení procesoru 8086:

- vynulují se příznaky IF a TF
- **provede se instrukce OUT**
- do zásobníku se uloží registr CS
- do zásobníku se uloží registr IP

Pro tabulku adres rutin obsluhujících přerušení procesoru 8086 neplatí

- začíná na adrese 0:0000
- začíná na začátku adresového prostoru
- má 256 řádků
- **začíná na adrese SS:0000**

Při přerušení v procesoru 8086 se jako první operace provádí

- **do zásobníku se uloží registr příznaků (F)**
- vynulují se příznaky IF a TF
- registr IP se naplní 16bitovým obsahem adresy $n \times 4$
- registr CS se naplní 16bitovým obsahem adresy $n \times 4 + 2$

Při návratu z přerušení v procesoru 8086 se provádí instrukce IRET, pro níž neplatí

- ze zásobníku se obnoví registr IP
- ze zásobníku se obnoví registr CS
- ze zásobníku se obnoví příznakový registr
- **ze zásobníku se obnoví registr SS**

Návrat do přerušeného procesu v procesoru 8086 typicky zajistí instrukce

- **IRET**
- MOV
- OUT
- POP

Mezi rezervovaná přerušení procesoru 8086 nepatří

- pokus o dělení nulou
- krokovací režim
- ladící bod
- **časovač**

Pro trasovací režim procesoru 8086 neplatí

- po provedení instrukce je generováno přerušení INT 1
- procesor je uveden do krokovacího režimu příznakem TF (Trace Flag)
- krokovací režim využívá instrukci IRET
- **probíhá, když je TF nastaven na nulu**

Příznak TF procesoru 8086 se nastaví na jedničku **Který ze zápisů instrukcí procesoru 8086 není korektní operací?**

- **při obnově příznakového registru (F) ze zásobníku instrukcí IRET**
- instrukcí SETTF
- při obnově registru IP ze zásobníku instrukcí IRET
- žádná z uvedených možností

- MOV AX,BX
- MOV AX,[BX]
- MOV AX,PROM[BX][DI]
- **žádná z uvedených možností**

Pro instrukci MOV procesoru 8086 neplatí

Signál RESET procesoru 8086 neprovede

- vynuluje IP
- vynuluje příznakový registr
- **nastaví TF = 1**
- vynuluje SS

- **mění příznaky**
- nelze s ní měnit registr CS
- má povolen tvar MOV BX,CX
- nemá povolen tvar MOV adresa,adresa

Chci naplnit registr AH procesoru 8086 hodnotou 50, které řešení není správné

- MOV AH,50
- PADESAT DB 50
- AH,PADESAT
- **MOV AH,[50]**

Který ze zápisů instrukcí procesoru 8086 je špatně

- **MOV CS,DS**
- MOV DS,adresa
- MOV adresa,DS
- MOV CX,DX

Chci naplnit registr AH procesoru 8086 obsahem adresy 50, které řešení je správné

- MOV AH,50
- PADESAT DB 50
- AH,50
- **MOV AH,[50]**
- žádná z uvedených možností

Pro instrukci procesoru 8086 MOV SS,... platí

- **po dobu trvání následující instrukce je zakázáno přerušení**
- po dobu trvání předchozí instrukce bylo zakázáno přerušení
- je nepovolená operace

Instrukce procesoru 8086 MOV AH,[BX] provede

- hodnota registru BX se uloží do registru AH
- **hodnota, která je na adrese v registru BX, se uloží do AH**
- registr BX se naplní hodnotou z adresy uložené v registru AH
- hodnota registru AH se uloží do registru BX

Pro aritmetické instrukce procesoru 8086 platí

- nesmí nastavovat příznaky
- nepatří sem instrukce ADD
- nepatří sem instrukce INC
- **žádná z uvedených možností**

Instrukce procesoru 8086 MOV AH,[BX][DI] provede

- hodnota vzniklá sečtením obsahů registrů BX a DI se uloží do registru AH
- **hodnota, která je na adrese, jež vznikne součtem adres v registrech BX a DI se uloží do AH**
- hodnota, která je uložena v AH se uloží do registrů BX a DI
- hodnota, která je na adrese, jež vznikne rozdílem adres v registrech BX a

Pro znaménkové rozšíření procesoru 8086 neplatí

- do všech bitů nové horní poloviny se zkopíruje znaménkový bit původního objektu
- znaménko je zachováno
- **všechny bity původního objektu se zkopírují do jeho nové horní poloviny**

Instrukce procesoru 8086 ADC se používá

- **při sčítání širších objektů**
- při násobení dvou čísel
- při odčítání s výpůjčkou
- při přičítání k obsahu registru CX

Při násobení reálných čísel procesoru 8086 použijeme instrukci

- IMUL
- MUL
- IDIV
- **žádná z uvedených možností**

Který z následujících skoků procesoru 8086 mění registr CS?

- **vzdálený (far)**
- krátký (short)
- blízký (near)
- žádná z uvedených možností

Který skok procesoru 8086 pracuje se 16bitovým operandem?

- vzdálený (far)
- krátký (short)
- **blízký (near)**
- žádná z uvedených možností

Který skok procesoru 8086 pracuje s 8bitovým operandem?

- vzdálený (far)
- **krátký (short)**
- blízký (near)
- žádná z uvedených možností

Pro podmíněný skok procesoru 8086 neplatí

- je vždy krátký
- reaguje na obsah příznaků
- **vždy mění registr CS**
- cílová adresa se vytvoří 8bitovým přírůstkem

Do zásobníku procesoru 8086 se vkládají

- 8bitové objekty
- **16bitové objekty**
- 32bitové objekty
- 64bitové objekty

Pro instrukci POP SS,... platí

- **po dobu trvání následující instrukce je zakázáno přerušení**
- po dobu trvání předchozí instrukce bylo zakázáno přerušení
- je nepovolená operace

Jaký je rozdíl mezi instrukcí RET a RETF procesoru 8086

- **RETF naplní i registr CS**
- RET naplní i registr CS
- RET i RETF pracují s 32bit. objekty, ale pouze RETF naplňuje registr CS
- RET i RETF pracují s 32bit. objekty, ale pouze RET naplňuje registr CS

Instrukce HALT procesoru 8086

- vynuluje registry
- vypne počítač
- **uvede procesor do stavu čekání**
- vynuluje příznaky a registry

Proč instrukce STI procesoru 8086 nepovoluje přerušení ihned?

- aby mohlo být provedeno instrukcí MOV SP,... atomické naplnění ukazatele vrcholu zásobníku
- aby mohl být nepřerušitelně zastaven procesor instrukcí HLT
- aby mohl být atomicky nepřerušitelně uložen ukazatel vrcholu zásobníku
- **aby byla nepřerušitelně ze zásobníku vybrána adresa přerušené instrukce**

Kde začíná segment reálného režimu (procesoru 8086)?

- na libovolné adrese
- na adrese dělitelné 4
- **na adrese dělitelné 16**
- na adrese dělitelné 32

Adresa reálného režimu procesorů Intel x86 ve tvaru segment : offset 01B2:0015 představuje dvacetibitovou adresu (hexadecimálně)

- **01B35**
- 01B17
- 011B5
- 002B5

Registr IP procesoru Intel 8086 obsahuje

- segmentovou část adresy právě prováděné instrukce
- **offsetovou část adresy právě prováděné instrukce**
- segmentovou část adresy následující instrukce
- offsetovou část adresy následující instrukce

Všechny odkazy na zásobník v procesoru Intel 8086 jsou segmentovány přes registr

- AX
- F
- **SS**
- CS
- DS

Instrukce IRET reálného režimu procesoru Intel x86 zajišťuje

- **návrat do přerušeného procesu a jeho pokračování**
- přerušení procesu
- přerušení procesu po provedení následující instrukce
- návrat od přerušeného procesu. Jeho pokračování zajistí jiná instrukce
- návrat z podprogramu

Je-li v procesoru 8086 nastaven příznak OF=1 a následně je provedena instrukce INTO, nastane

- přepnutí do krokovacího režimu
- **přerušení INT 4**
- návrat do přerušeného procesu
- obnovení registru IP ze zásobníku

Procesor 8086 poskytuje pro adresování V/V bran

- 8bitovou adresu
- **16bitovou adresu**
- 20bitovou adresu
- 24bitovou adresu

Která z následujících operací procesoru Intel x86 je nekorektní

- MOV BX,CX
- MOV DI,10000
- MOV AX,CS
- **MOV CS,AX**

Instrukce AND Intel x86 provádí

- **logický součin**
- logický součet
- přidání hodnoty ze zdrojového registru do cílového
- součet dvou čísel
- přičtení čísla

Instrukce IN AX,DX procesoru x86 zajišťuje

- přenos slabiky z AL do V/V brány podle DX
- přenos slova z AL do V/V brány podle DX
- přenos slabiky z V/V brány podle DX do registru AX
- **přenos slova z V/V brány podle DX do registru AX**

Instrukce INC CL provede v procesoru x86

- $CL := CL + CL$
- **$CL := CL + 1$**
- $CL := 1$
- $CL := 0$

Kolika bitovou adresu při přístupu do paměti vytváří procesor Intel 8086?

- 16
- **20**
- 24
- 32

Adresa SS:SP ukazuje vždy na

- dno zásobníku
- **na vrchol zásobníku**
- na adresu naposled prováděné instrukce
- na adresu naposled použité V/V brány

Jaká je maximální dosažitelná adresa v reálném režimu procesoru Intel 80286 a vyšších procesorů

- 0FFFFh
- **10FFEFh**
- 10FFFFh
- 10FFFEh

Kolik řádků má tabulka popisovačů segmentů GDT nebo LDT procesoru Intel 80286 a vyšších procesorů

- **8192**
- 4096
- 16384
- 65536

Virtuální adresa procesoru Intel 80286 má celkem 30 bitů na adresaci virtuální paměti. Jak velká tato virtuální paměť může být?

- **1 GB**
- 4 GB
- 2 GB
- 16 MB

S obsahem instrukčního segmentu procesoru Intel 80286 je povoleno následující:

- číst a provádět; mám-li potřebná práva, pak i zapisovat
- **pouze provádět a možná i číst; mám-li potřebná práva**
- cokoli, pokud mám potřebná práva

Popisovač segmentu s LDT (tabulka popisovačů lokálního adresového prostoru) se v procesoru Intel 80286 smí nacházet v těchto tabulkách

- **pouze v GDT**
- v GDT i v LDT
- v GDT a IDT
- v žádné z nich

Popisovač segmentu s GDT (tabulka popisovačů globálního adresového prostoru) se v procesoru Intel 80286 smí nacházet v těchto tabulkách

- pouze v GDT
- v GDT i v LDT
- pouze v IDT
- **v žádné z nich**

V reálném režimu procesoru Intel 80286 nelze provést instrukci

- **LLDT (plnění registru LDTR)**
- LGDT (plnění registru GDTR)
- LIDT (plnění registru IDTR)
- LMSW (plnění registru MSW)
- HLT (zastavení procesoru)

Jaký je rozdíl mezi přerušením typu trap a fault v procesoru Intel 80286?

- Fault je fatální stav, ze kterého se nelze zotavit. Z přerušení trap se zotavit lze.
- **Fault pracuje s adresou ukazující na instrukci, která přerušení způsobila. Trap poskytuje adresu ukazující na instrukci následující.**
- Přerušení typu trap je obsluhováno branou z tabulky IDT a přerušení fault

je obsluhováno branou z tabulky GDT.

Co znamená výjimka (přerušení) "Výpadek segmentu" v procesoru Intel 80286?

- **procesor při vyčíslování virtuální adresy narazil na nulovou hodnotu bitu Present**
- procesor při vyčíslování virtuální adresy narazil na nulovou hodnotu bitu Accessed
- procesoru se nepodařilo vyčíslit reálnou adresu z virtuální

Procesor 80286 má

- **16bit. datovou a 24bit. adresovou sběrnici**
- 24bit. datovou a 20bit. adresovou sběrnici
- 32bit. datovou a 24bit. adresovou sběrnici
- 24bit. datovou a 32bit. adresovou sběrnici

Procesor 80286 má

- **chráněný a reálný režim**
- chráněný a virtuální režim
- sběrniceový a reálný režim
- reálný a nereálný režim

Pro chráněný režim procesoru 80286 neplatí

- není možnost jej softwarově vypnout
- **tabulka vektorů přerušení má velikost 1 KB**
- poskytuje prostředky 4úrovňové ochrany
- adresuje 16 MB reálné paměti

Pro registr MSW procesoru 80286 neplatí

- slouží k zapnutí chráněného režimu
- **slouží k zapnutí reálného režimu**
- plní se instrukcí LMSW
- čte se instrukcí SMSW

Signál RESET u procesoru 80286

- zapíná chráněný režim procesoru
- **zapíná reálný režim procesoru**
- vypíná koprocessor
- žádná z uvedených možností

Bit P popisovače datového segmentu procesoru 80286 nastavený na 1 sděluje:

- obsah segmentu je uložen na disku
- obsah segmentu je prázdný
- **obsah segmentu je uložen v reálné paměti**
- je vždy automaticky nastaven na jedničku

Bit ED datového segmentu procesoru 80286 určuje

- **zda datový segment obsahuje zásobník**
- přistupová práva k segmentu
- zakazuje čtení obsahu segmentu
- zakazuje zápis do segmentu

Bit C (Conforming) popisovače

instrukčního segmentu procesoru 80286 rezervovaná přerušení 80286?

- **může způsobit změnu úrovně oprávnění pro podprogramy volané v tomto segmentu**
- indikuje směr rozšiřování segmentu
- je nastaven na jedna, je-li procesor v reálném režimu
- žádná z uvedených možností

Pro registr GDTR procesoru 80286 neplatí

- má délku 5 bajtů
- při spuštění chráněného režimu se do něj vkládá adresa tabulky GDT
- naplňuje se instrukcí LGDT
- **označuje segment stavu procesoru**

Pro TSS (segment stavu procesoru 80286) neplatí

- na segment TSS ukazuje popisovač systémového segmentu, který může být umístěn pouze v GDT
- slouží k uložení kontextu procesu, kterému bylo odebráno řízení
- **je to ukazatel, jestli je procesor 80286 v chráněném režimu**
- každý proces má vlastní TSS

Interrupt Descriptor Table (IDT) procesoru 80286 nemá popisovač

- brána zpřístupňující TSS
- brána pro maskující přerušení
- brána pro nemaskující přerušení
- **brána pro V/V operace**

Pro Interrupt Descriptor Table (IDT) procesoru 80286 neplatí

- obsahuje až 256 popisovačů rutin obsluhujících přerušení
- její adresu obsahuje IDTR
- **slouží k uložení kontextu procesu, kterému bylo odebráno řízení**
- obsahuje nejvýše tolik popisovačů, kolik dovoluje limit segmentu

Který z následujících názvů nspecifikuje kategorii přerušení generovanou procesorem 80286?

- Fault
- Trap
- Abort
- **Flag**

Která z možností nepatří mezi rezervovanou přerušení 80286?

- dělení nulou
- přeplnění
- chybný operační kód
- **výpadek systému**

Zapnutí chráněného režimu procesoru 80286 neznamena

- změnu způsobu adresace
- nastavení bitu PE=1 registru MSW
- vypnutí reálného režimu
- **restart procesoru**

Počet lokálních adresových prostorů procesoru 80286 typicky se rovná

- **počtu spuštěných procesů**
- počtu uplatněných přerušení
- jedné
- záleží na operačním systému
- velikosti tabulky GDT

Procesor Intel 80386 je

- **32bitový procesor s 32bitovou adresovou a datovou sběrnici**
- 32bitový procesor s 24bitovou vnější a 32bitovou vnitřní adresovou sběrnici
- 32bitový procesor s 24bitovou adresovou a 32bitovou datovou sběrnici

Selektor v chráněném režimu procesoru Intel 80386 je

- **16bitový**
- 32bitový
- 48bitový
- 64bitový

Procesor Intel 80386 pracuje s těmito možnými adresami

- **48bitovou logickou adresou, 32bitovou lineární adresou a 32bitovou fyzickou adresou**
- 64bitovou logickou adresou, 48bitovou lineární adresou a 32bitovou fyzickou adresou
- 48bitovou logickou adresou, 48bitovou lineární adresou a 32bitovou fyzickou adresou

Kolika bity plní programátor segmentové registry v procesoru Intel 80386 a vyšších typech?

- **16**
- 32
- 48
- 64

Stránkováním se v procesoru Intel 80386 transformuje

- logická adresa na lineární
- fyzická adresa na lineární
- **lineární adresa na fyzickou**
- logická adresa na fyzickou

Největší možná velikost segmentu v procesoru Intel 80386 a vyšších typech je

- 64 KB
- 1 MB
- 4 MB
- 1 GB
- **4 GB**

Velikost stránky v procesoru Intel 80386 a vyšších typech je

- maximálně 4 KB
- **právě 4 KB**
- maximálně 1 KB
- právě 1 KB

Co znamená "Mapa přístupných V/V bran" v procesoru Intel 80386?

- Seznam existujících V/V adres na počítači.
- **Seznam V/V adres dostupných jednomu konkrétnímu (typicky V86) procesu.**
- Seznam V/V adres dostupných (typicky V86) procesům chráněného režimu.

Jaká část adresy vstupující do stránkovací jednotky není stránkováním postihnuta (v procesoru Intel 80386)?

- **dolních 12 bitů**
- dolních 10 bitů
- horních 10 bitů
- horních 20 bitů

Kolik bitů je nezbytných pro uložení adresy stránkovací tabulky (zpravidla ve stránkovacím adresáři) a stránkovacího adresáře (zpravidla v CR3)?

- **20**
- 32
- 16

Pro procesor 80386 neplatí

- datová sběrnice má 32 bitů
- lze použít stránkování
- data se do/z paměti přenášejí po 4 bajtech
- **adresová sběrnice má 24 bitů**

Pro adresaci v chráněném režimu procesoru 80386 neplatí

- **offset je 16bitový**
- selektor je stejný jako v 80286
- báze segmentu je 32bitová
- limit segmentu může být až 4 GB

Co znamená, že stránkovací jednotka procesoru 80386 není zapnuta

- **fyzická adresa je totožná s lineární adresou**
- fyzická adresa obsahuje 48 bitů
- lineární adresa obsahuje 48 bitů
- lineární adresa je totožná s logickou adresou

Pro stránkování procesoru 80386 platí

- je povinně zapnuto
- je-li vypnuto, tak se lineární adresa transformuje na logickou
- **je-li zapnuto, tak se lineární adresa transformuje na fyzickou**
- žádná z uvedených možností

Pro stránkový adresář procesoru 80386 neplatí

- má velikost právě jedné stránky
- ukazuje na max. 1024 stránkových tabulek
- je k dispozici pouze se zapnutým stránkováním
- **žádná z uvedených možností**

Pro bit D (Dirty) při stránkování procesoru 80386 neplatí

- procesor ho nastaví na jedničku při zápisu do rámce
- procesor jej nenuluje, to má na starost software
- rozlišuje, jestli je rámec špinavý nebo čistý
- **pokud je nastaven na jedničku, tak je rámec vybrán za oběť**

Pro TLB neplatí

- funguje na principu asociativní paměti
- **je zapnuto pouze v chráněném režimu procesoru 80286**
- je to vyrovnávací paměť
- při vyprazdňování se vynulují bity V (validita)

Procesor 80386 má 32 bitovou adresovou sběrnici A2 až A31, což znamená, že

- do paměti se jde alespoň pro 1 bajt
 - do paměti se jde alespoň pro 2 bajty
 - **do paměti se jde alespoň pro 4 bajty**
 - do paměti se jde alespoň pro 8 bajtů
-

Jaký má význam interní vyrovnávací paměť v procesoru Intel 80486?

- Pamatuje si posledních několik transformovaných lineárních adres na fyzické.
- **Pamatuje si několik posledních obsahů adres čtených z fyzické paměti vč. okolí.**
- Vyrovnává rozdíly toku dat mezi interními jednotkami procesoru pro proudové zpracování (pipeline).

Kolik bitů by potřeboval algoritmus LRU v interní vyrovnávací paměti procesoru Intel 80486 k tomu, aby úplně fungoval pro výběr ze čtyř položek na řádku (předpokládejme, že by byl realizován neúplnou maticí)?

- 3
- 4
- **6**
- 8
- 10

Procesor 80486 nemá

- datovou sběrnici 32bitů
- adresovou sběrnici 32bitů
- integrovaný matematický koprocessor
- **žádná z uvedených možností**

Procesor 80486 se od procesoru 80386 neliší v

- **velikosti vnějších sběrnic**
- interní vyrovnávací paměti
- nové technologii, která se blíží k RISCovým procesorům
- jednotce operací v pohyblivé řádové čárce

V procesoru Pentium

- se dynamicky předvídá výskyt nepodmíněné skokové instrukce
 - se dynamicky předvídá výskyt podmíněné skokové instrukce
 - **se dynamicky předvídá výsledek vyhodnocení podmínky podmíněné skokové instrukce**
 - se staticky předvídají nepodmíněné skoky dvouvariantními instrukcemi
-

Který rys je vlastní technologii procesorů RISC?

- usnadnění programování pro člověka programátora
- zrychlení provádění poskytnutím co nejbohatších instrukcí
- integrování vnější paměti dovnitř procesoru
- **poskytnutí velkého počtu registrů v procesoru**

Základní šířka dat interně zpracovávaných koprocесorem pro výpočty v pohyblivé řádové čárce je

- **80 bitů**
- 128 bitů
- 64 bitů
- 40 bitů
- 32 bitů

Nejmenší záporné číslo (největší v absolutní hodnotě; číslo na levé hranici rozsahu zobrazení) v IEEE 754 má

- **znaménko mantisy 1, největší kladné zobrazitelné číslo v exponentu.**
- znaménko mantisy 1, nejmenší záporné zobrazitelné číslo v exponentu.
- znaménko mantisy 1, nulový exponent.
- znaménko mantisy 0.

Signály STROBE a BUSY používá rozhraní

- RS-232
- V.24
- **Centronics**
- USB

Paralelní rozhraní je

- RS-232.
- **Centronics.**

Rozhraní Centronics: Signál !STROBE je v aktivní úrovni

- dokud neuplyne doba "předstih"
- dokud neuplyne doba "přesah"
- dokud tiskárna signálem BUSY neoznámí konec zpracování
- **pevně stanovenou dobu**

Rozhraní Centronics: Signál !STROBE je v aktivní úrovni

- **když přenáší hodnotu logická "0"**
- když přenáší hodnotu logická "1"

Rozhraní RS-232C: Přenos dat tímto rozhraním je:

- synchronní
- asynchronní
- **synchronní i asynchronní**
- nic z toho

Rozhraní RS-232C: Jaké zapojení nulmodemu je nesmyslné?

- SG--SG, TxD--RxD, RxD--TxD
- **SG--SG, TxD--TxD, RxD--RxD**
- SG--SG, TxD--RxD, RxD--TxD, RTS+CTS--DCD, DCD--RTS+CTS

USB při komunikaci používá protokol

- **Master-Slave**
- CSMA/CD
- Token-Ring