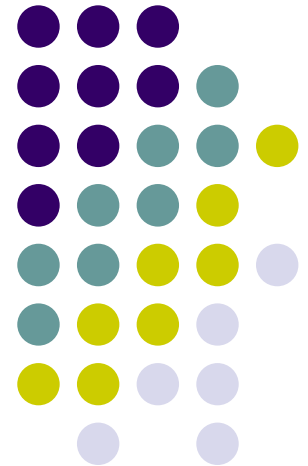


První mikroprocesory





Mikroprocesor

- Mikroprocesor je složitý **integrovaný obvod**, který dokáže provádět zadané **instrukce**
- Mikroprocesor je **sekvenční logický** obvod
- Mikroprocesor zpracovává **data v číslíkové podobě**
- Data jsou zpracovávána dle programu, jehož povely jsou zakódovány ve strojovém kódu
- Mikroprocesory pracují pouze s **digitálním signálem**

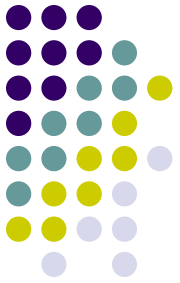


Instrukční sada

- Každý mikroprocesor má svou **instrukční sadu**
- Instrukční sada = **repertoár instrukcí**, které dokáže daný mikroprocesor vykonávat
- Každý typ mikroprocesoru má jinou instrukční sadu
- Pokud dva různé mikroprocesory mají stejnou instrukční sadu a strojový kód, jsou **kompatibilní** – stejný program ve strojovém kódu půjde spustit na obou procesorech
- Mikroprocesory mohou být **zpětně kompatibilní**, pokud instrukční sada novějšího mikroprocesoru obsahuje všechny instrukce z instrukční sady nějakého staršího typu
- **Program** je posloupnost instrukcí
- Program pro daný mikroprocesor může obsahovat pouze instrukce z jeho instrukční sady
- Instrukční sada může být různě široká
 - Existují mikroprocesory s bohatou instrukční sadou (např. 500 a více instrukcí)
 - Existují mikroprocesory s malou instrukční sadou (např. pouze kolem 20 instrukcí)

Strojový kód

- JK , ukázka





Počátky mikroprocesorů

- Počínaje třetí generací počítačů se začínají objevovat první mikroprocesory
- Mikroprocesor vzniká splynutím **řadiče** a **ALU** do jediného funkčního bloku integrovaného na jediném čipu
- Základním parametrem procesoru je **počet bitů**, tzn. šířka dat, která je procesor schopen naráz zpracovat v jednom kroku
- Instrukce **osmibitového** procesoru umí pracovat s jedním bajtem (například lze inkrementovat osmibitové číslo)
- Instrukce **šestnáctibitového** procesoru již zpracují 16 bitů naráz
- Osmibitový procesor má osmibitové registry, šestnáctibitový procesor má šestnáctibitové registry atd....
- **Registr** je velmi malá, ale velmi rychlá paměť nacházející se v přímo v mikroprocesoru
- **Registry** mají své **jméno** , např AX, CS, R2 ... (narozdíl od paměťových míst, která jsou identifikována svou adresou)

Registry



- Registry slouží zejména k **dočasnému uložení operandů**, se kterými právě pracuje
- Mnoho instrukcí obvykle umí pracovat pouze s bajty uloženými v registrech
- Běžné procesory neumí sečíst dva bajty ležící v paměti, ale umí sčítat bajty ležící v registrech procesoru - je nutné nejprve sčítané bajty přesunout z paměti do registrů
- Registrů je obvykle v procesoru poměrně **malý počet**
- Bajty, které není nutné mít okamžitě k dispozici k budoucím operacím se obvykle zbytečně neuchovávají v registrech, ale odkládají se do paměti
- **Paměť** tedy slouží k **dlouhodobému uložení dat**
- S daty v paměti ale nelze provádět všechny operace
- Registry slouží k uložení dat, se kterými se právě pracuje a která v registrech být musí, protože daná operace vyžaduje uložení operandů v registru

Registry



- Například mikroprocesor i8051 disponuje pracovními registry **A**, **B**, **R0**, **R1**, **R2**... **R7** (kromě dalších speciálních řídících registrů)
- Instrukce násobení *MUL* dokáže vynásobit pouze bajty ležící v registrech **A** a **B** a výsledný celočíselný součin uloží automaticky do registru **A**
- Chceme-li vynásobit bajt uložený v paměti na adrese **50** bajtem v registru **R2**, je nejprve potřeba přesunout tyto dva bajty do registrů **A** a **B**



Střadačová architektura

- U některých mikroprocesorů je běžné, že některé instrukce lze provádět pouze s některými konkrétními registry
- Výsadní postavení má registr zvaný **střadač** (označovaný obvykle A, ACC, AX apod.), se kterým lze provádět všechny operace
- Do střadače se také obvykle automaticky ukládají všechny výsledky
- S ostatními registry lze provádět pouze omezený okruh operací
- S daty v paměti pak obvykle nelze provádět přímo nic kromě čtení/zápisu (tzn. přesunu do/z registru)
- Taková architektura procesu se nazývá **střadačová**
- **Střadačovou architekturu** mají například procesory rodiny Intel 8051 nebo 8086



Univerzální registry

- Procesor obsahuje několik registrů, které mají naprosto rovnoprávné postavení
- Architektura s univerzálními registry nepoužívá privilegovaný střadač
- Veškeré operace lze provádět s **kterýmkoliv registrem**
- Výsledky lze ukládat do libovolného registru
- Zatímco na střadačové architektuře musí programátor ve strojovém kódu neustále přemýšlet, jak správně rozmístit do registrů zpracovávaná data, aby prováděné instrukce uměli právě s těmito registry pracovat, na architektuře s univerzálními registry tato komplikace odpadá
- Tato architektura je dnes typická například pro RISC procesory



Registry - shrnutí

- Registry slouží k **uložení okamžitých operandů**
- Registry jsou přímo součástí mikroprocesoru
- Každý registr je označen jménem
- Šířka registrů je dána počtem bitů procesoru (32-bitový procesor = 32 bitové registry)
- Data je obvykle nejprve třeba přesunout z paměti do registrů procesoru
- Registrů je málo (např. 10) a proto jsou v nich uloženy jen právě používané nutné datové bajty
- Výsledky nepotřebné k dalšímu okamžitému zpracování se odkládají k dlouhodobému uchování zpět do paměti

Šířka zpracovaných dat



- Úkolem je sečíst čísla 1A57h + 2362h

- **16-bitový mikroprocesor**

- Má 16-bitové registry a 16-bitovou ALU

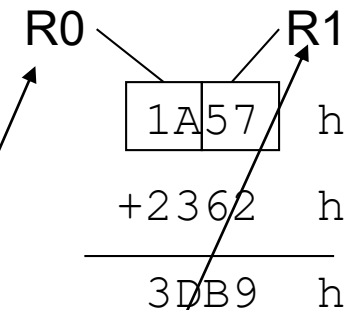
```
MOV R0, 1A57h    // do jednoho registru umístíme první sčítanec
MOV R1, 2362h    // do druhého registru umístíme druhý sčítanec
ADD R0, R1       // sečteme obsah obou registrů
```

- **8-bitový mikroprocesor**

- Má 8-bitové registry a 8-bitovou ALU
- Sčítaná čísla jsou 16-bitová, ale registry jsou pouze 8-bitové
- Sčítance bude nutné rozdělit na dvě osmibitové části a výsledek vypočítat po částech (jako když sčítáte dvě velká čísla pod sebou na papíře)

```
MOV R0, 1Ah // do jednoho registru umístíme horních osm bitů prvního sčítance
MOV R1, 57h // do tohoto registru umístíme spodních osm bitů prvního sčítance
MOV R2, 23h // do tohoto registru umístíme horních osm bitů druhého sčítance
MOV R3, 62h // do tohoto registru umístíme spodních osm bitů druhého sčítance
ADD R1, R3  // sčítáme spodních osm bitů
ADD R0, R2  // a nakonec horních osm bitů obou sčítanců
```

- Na osmibitovém mikroprocesoru lze sečíst dvě 16-bitová čísla, ale nikoliv naráz, je potřeba sčítat postupně po 8 bitech (pro jednoduchost v tomto příkladu ignorujeme možnost přetečení prvního výsledku. Ve skutečnosti by bylo navíc nutné po prvním provedeném sčítání testovat zda nepřeteklo a pokud ano, k výsledku druhého sčítání přičíst jedničku jako přenos)
- Podobně by bylo možné na 8-bitovém mikroprocesoru sečíst například dvě 64-bitové čísla. Evidentní je, že to ale bude mnohem víc práce, než by s tím měl 64-bitový mikroprocesor





- Co to znamená, když se řekne, že procesor pracuje s 8-bitovými daty a používá 16-bitové adresy ?
- Je takový procesor osmibitový nebo šestnáctibitový ?

Šířka adresy

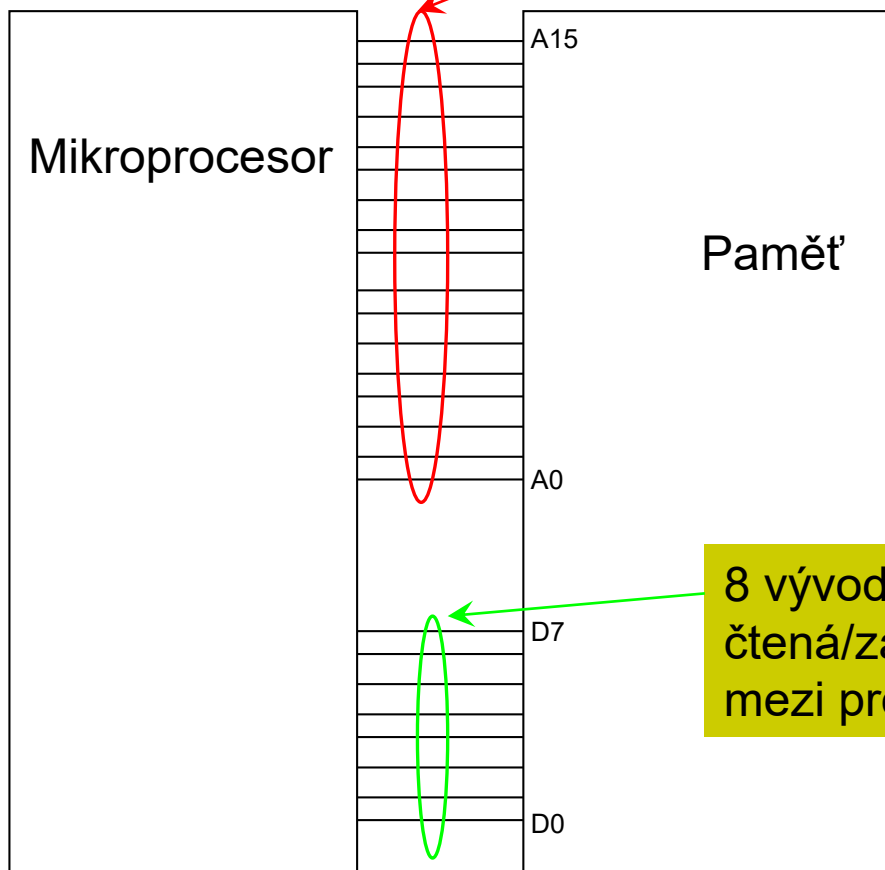


- Jednotlivá paměťová místa jsou identifikována svou adresou
- Kdyby osmibitový mikroprocesor používal osmibitové adresy, nemohl přistupovat výše než k adrese 255
- Procesor s osmibitovou adresací tedy může adresovat max. 256 B paměti
- Bývá poměrně běžné, že šířka adresace, kterou procesor používá se neshoduje s šířkou dat, se kterými procesor pracuje
- Například je běžné, že osmibitový procesor používá šestnáctibitovou adresaci
- Takový procesor má osmibitové registry a umí manipulovat pouze s osmi bity, ale dokáže přistupovat kamkoliv do 64 kB paměti, protože umí vygenerovat adresu 0 – 65535
- Takový procesor tedy například umí zapsat na adresu 32852 bajt s hodnotou 200, ale neumí počítat 300+500
- Setkáme s nejrůznějšími kombinacemi počtu bitů procesoru a šířkou adresace (např. šestnáctibitový procesor i8086 používá 20-bitovou adresaci)

Osmibitový mikroprocesor s šestnáctibitovou adresací



16 vývodů, kterými mikroprocesor vybírá jednu z 65536 adres, se kterou se bude pracovat



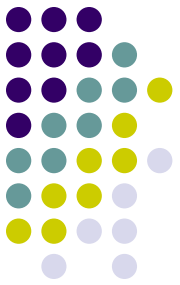
8 vývodů, kterými se přenáší čtená/zapisovaná data (8 bitů) mezi procesorem a pamětí

Speciální registry



- Kromě běžných registrů sloužících k uložení operandů mívají mikroprocesory další speciální registry se zvláštním určením
- U většiny mikroprocesorů se setkáme s těmito spec. registry
 - Řídící registry
 - Čítač instrukcí / programový čítač
 - Ukazatel na vrchol zásobníku
 - Příznakový registr

Řídící registry



- Skrze tyto registry lze **konfigurovat a ovládat** činnost mikroprocesoru a jeho funkčních jednotek
- Zápisem do příslušného řídícího registru tak lze například povolit/zakázat přerušení, zapnout režim se sníženým příkonem, přejít do jiného režimu činnosti (např. z reálného do chráněného u procesorů 286 a vyšších)....
- S těmito registry nelze provádět žádné aritmeticko-logické operace
- Každý konkrétní mikroprocesor disponuje jinou sadou speciálních řídících registru



Programový čítač

- Jeho účelem je adresovat instrukce strojového kódu v operační paměti počítače
- Z pohledu programátora by se zjednodušeně dalo říct, že ukazuje na „řádek“ programu, který se právě zpracovává
- Protože ve strojovém kódu ale žádné řádky neexistují a instrukce jsou zakódovány jako bajty strojového kódu, musí jít o ukazatel na konkrétní bajt stroj. kódu
- Tento registr tedy slouží jako „ukazovátko“ do paměti na místo, kde leží strojový kód právě zpracovávané instrukce
- Po načtení strojového kódu instrukce a jejím zpracování se automaticky zvyšuje tak, aby ukazoval na adresu, kde v paměti leží strojový kód následující instrukce
- Obvykle se nazývá anglicky
 - PC – Program Counter
 - IP – Instruction Pointer



Ukazatel vrcholu zásobníku

- Většina moderních mikroprocesorů si při své činnosti buduje v operační paměti tzv. zásobník
- Do zásobníku si procesor „na hromadu“ odkládá návratové adresy při volání podprogramu nebo přerušení
- ukazatel zásobníku je speciální registr, který udává adresu poslední přidané položky
- S činností zásobníku se detailněji seznámíme později
- Na většině mikroprocesorů bývá tento registr nazýván **SP** (Stack Pointer)



Příznakové bity

- Kromě registrů sloužících k uložení operandů bývá součástí mikroprocesoru speciální tzv. příznakový registr
- Jednotlivé bity tohoto registru fungují jako jakési indikátory zajímavých situací, které mohou nastat při provádění aritmeticko-logických operací
- Typicky jde o příznaky
 - Nulového výsledku (takový bit je nastaven, pokud výsledkem poslední operace je nula)
 - Záporného výsledku
 - Přetečení (Výsledek matematické operace je vyšší než je možno uložit , např. více než 255 v případě osmibitového procesoru)
 - Parity (Počet bitů s log. úrovní 1 ve střadači je sudý/lichý)
 - Přenosu mezi třetím a čtvrtým bitem (důležité pro počítání v BCD kódu)
- Každý konkrétní mikroprocesor disponuje jinou sadou příznaků
- Stav příznakových bitů lze testovat a dle výsledku větvit program
- (Například: který ze dvou bajtů je větší, zjistíme tak, že je odečteme a pak se testuje příznak záporného výsledku)

Taktovací frekvence

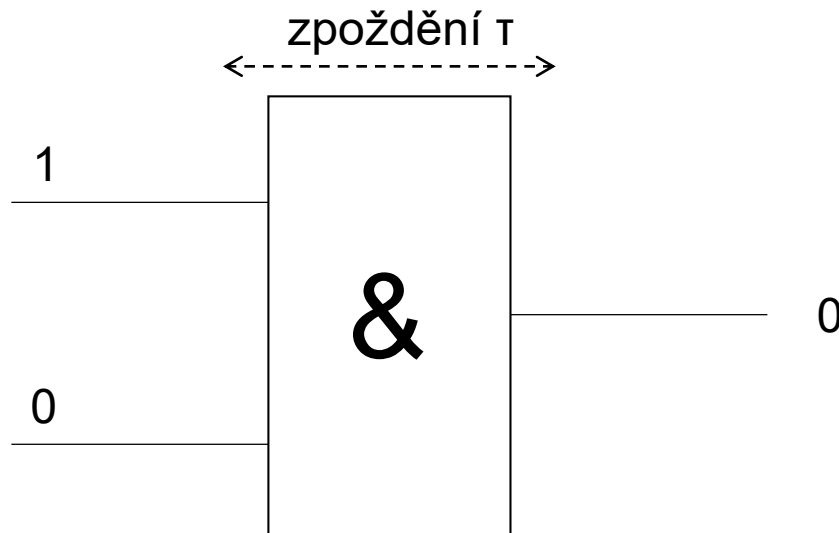


- Jedním z základních parametrů procesorů, který budeme sledovat, je **taktovací frekvence**
- Mikroprocesor je **synchronní sekvenční logický obvod**, který je synchronizován **hodinovým signálem**
- Čím vyšší taktovací frekvence, tím rychleji mikroprocesor pracuje
- Máme-li dva **stejně** mikroprocesory běžící na různé frekvenci, potom mikroprocesor pracující na vyšší frekvenci bude pracovat rychleji
- To ale neplatí pro dva **různé** mikroprocesory - výkonný mikroprocesor může na nižší frekvenci být rychlejší než slabý mikroprocesor běžící na vyšší frekvenci
- Taktovací frekvence tedy nic neříká o **skutečném výpočetním výkonu** mikroprocesoru a výkon různých mikroprocesorů nelze poměřovat poměrem jejich taktovacích frekvencí

Zpoždění log. obvodu



- Po přivedení signálu na vstupy logického obvodu se na jeho výstupu objeví výsledek
- Výstupní logická úroveň se na výstupu nenastaví ihned, ale až po určitém krátkém **zpoždění**
- Typické zpoždění běžných logických obvodů se pohybuje v řádu jednotek nanosekund



Zpoždění uvnitř mikroprocesoru



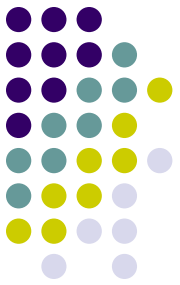
- **Mikroprocesor** je složen z tisíců takových logických obvodů, jejichž zpoždění se sčítá
- Na vstup mikroprocesoru přivádíme **povely** zakódované ve **strojovém kódu**
- Mikroprocesor není schopen vykonat zadané povely **ihned** v nekonečně krátkém čase, ale vlivem **zpoždění** v logických obvodech, ze kterých je složen, mu to určitou dobu trvá
- Další povel může mikroprocesor vykonávat až poté, co **dokončil předchozí**
- Mezi prováděním jednotlivých instrukcí tedy musí uplynout určitý čas
- Kdyby mikroprocesor začal provádět další instrukci dříve, než dokončí výpočet instrukci předchozí, docházelo by k chybám – následující instrukce by například mohla chtít pracovat s výsledkem předchozí, který se ovšem nestihl vypočítat

Synchronizace

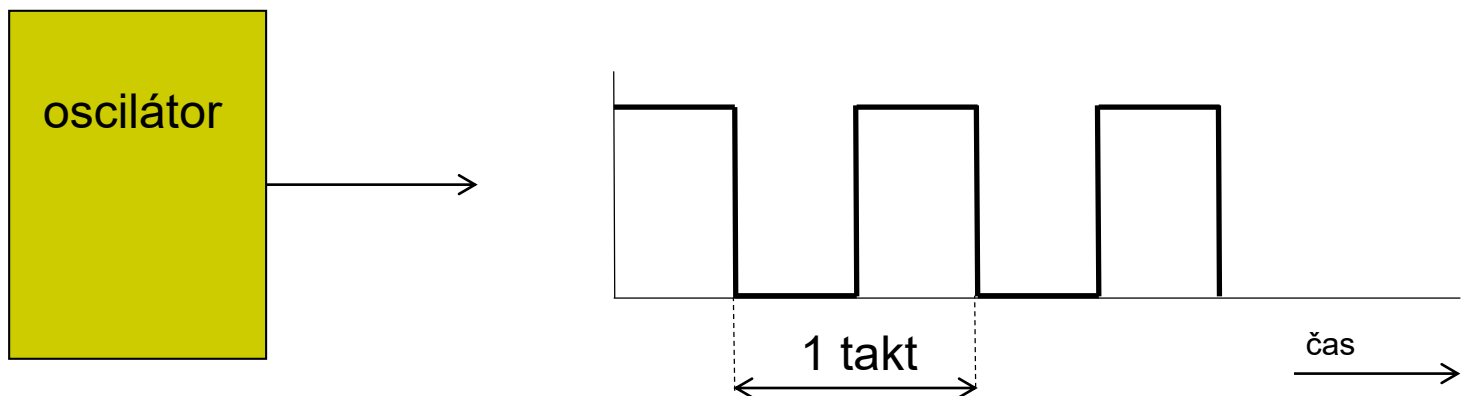


- Situace je ještě trochu komplikovanější
- Mikroprocesor nevykonává zadaný povel (instrukci) naráz, ale postupně v několika **fázích**
- Vzájemná **koordinace** činnosti jednotlivých funkčních bloků je zajištěna **řadičem**, který jim ve správný časový okamžik říká **řídícím signálem**, co mají dělat
- **Následující fáze** může být prováděna až poté, co je dokončena **fáze předchozí**
- např. zápis výsledku do paměti může být proveden až poté, co byl nejdříve tento výsledek vypočítán
- Pokud by řadič dal povel k zápisu výsledku do paměti dříve, než ho ALU vypočítá, zapsal by se vlastně do paměti nesmysl
- Řadič tedy musí mít „ponětí“ o čase a provádění jednotlivých fází zahajovat ve správný okamžik, s patřičným časovým odstupem
- Řadič nutně potřebuje **hodinový signál**
- Tímto hodinovým signálem je **synchronizována** veškerá činnost

Oscilátor



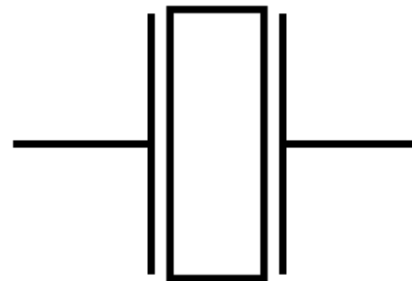
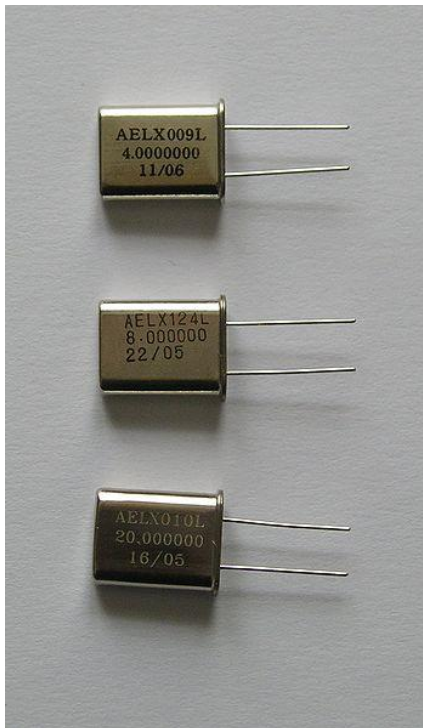
- **Oscilátor** generuje **hodinový signál**
- Hodinový signál je signál s **obdélníkovým průběhem** a **přesnou konstantní frekvencí**
- Od periody tohoto hodinového signálu jsou odvozeny veškeré časové poměry
- Perioda hodinového signálu se také nazývá **takt**
- Oscilátor je samostatný obvod, který nebývá součástí mikroprocesoru



Krystal



- Pokud by při konstrukci oscilátoru byly použity klasické součástky (rezistory, kondenzátory, cívky) byla by generovaná frekvence nepřesná (záleželo by na přesnosti součástek) a nestálá (záleželo by hodně na okolních vlivech (teplota, vlhkost...))
- **Krystalový oscilátor** - Přesnost generované frekvence je dosažena pomocí krystalu
- Krystal je součástka, která se chová podobně jako rezonanční obvod



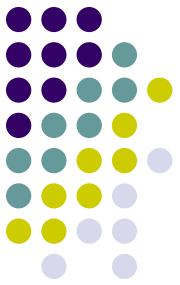
Frekvence hodinového signálu



- Jednotlivé fáze, na které je rozděleno provádění instrukce, obvykle netrvají stejně dlouho
- Ukažme si příklad na instrukci INC 100
- Tato instrukce má inkrementovat hodnotu bajtu, který leží v paměti na adrese 100
- Provádění této instrukce bude rozděleno do několika fází:
 - Přečtení strojového kódu instrukce z paměti (mikroprocesor musí zjistit, jakou instrukci má vlastně provádět)
 - Řadič generuje řídicí signál pro paměť, kterým vybírá adresu, ze které je potřeba přečíst strojový kód a dalším řídicím signálem sděluje, že z paměti se bude číst
 - Potřebný čas na přečtení strojového kódu z paměti: **25 ns**
 - Strojový kód je uvnitř mikroprocesoru dekodován a řadič plánuje provedení instrukce
 - Potřebný čas na dekodování přečteného strojového kódu: **12 ns**
 - Přečtení bajtu, který má být inkrementován, z paměti z adresy 100
 - Řadič generuje řídicí signál, kterým vybírá adresu 100 a dalším řídicím signálem paměti sděluje, že z této adresy chce číst
 - Potřebný čas na přečtení datového bajtu z paměti: **25 ns**
 - Výpočet – Inkrementace
 - Řadič generuje řídicí signál, který říká ALU, aby inkrementovala zadaný bajt
 - ALU provádí výpočet (časově náročné)
 - Potřebný čas na provedení výpočtu: **63 ns**
 - Zápis výsledku zpět do paměti (inkrementovaný bajt se uloží zpět na adresu 100)
 - Řadič generuje řídicí signál, kterým vybírá adresu 100 a dalším řídicím signálem sděluje, že na tuto adresu se bude zapisovat
 - Zápis bajtu do paměti trvá **36 ns**
 - Dokončení instrukce (nastavení příznaku přetečení, nuly, úklid po provedené práci, návrat do počátečního stavu....)
 - Doplňkové akce spojené s dokončením instrukce trvají **19 ns**

Řadič musí mezi generováním jednotlivých řídicích signálů dodržet dostatečný časový odstup.

...inového signálu



- Jednotlivé fáze, na které je rozděleno provádění instrukce, obvykle netrvají stejně dlouho
- Ukažme si příklad na instrukci INC 100
- Tato instrukce má inkrementovat hodnotu bajtu, který leží v paměti
- Provádění této instrukce bude rozděleno do několika fází:

- Přechtení strojového kódu instrukce z paměti (mikroprocesor musí z
- Řadič generuje řídicí signál pro paměť, kterým vybírá adresu, ze kt
- sděluje, že z paměti se bude číst
- Potřebný čas na přechtení strojového kódu z paměti: **25 ns**

- Strojový kód je uvnitř mikroprocesoru dekodován a řadič plánuje pr
- Potřebný čas na dekodování přechteného strojového kódu: **12 ns**

- Přechtení bajtu, který má být inkrementován, z paměti z adresy 100
- Řadič generuje řídicí signál, kterým vybírá adresu 100 a dalším říd
- Potřebný čas na přechtení datového bajtu z paměti: **25 ns**

- Výpočet – Inkrementace
- Řadič generuje řídicí signál, který říká ALU, aby inkrementovala za
- ALU provádí výpočet (časově náročné)
- Potřebný čas na provedení výpočtu: **63 ns**

- Zápis výsledku zpět do paměti (inkrementovaný bajt se uloží zpět r
- Řadič generuje řídicí signál, kterým vybírá adresu 100 a dalším říd
- Zápis bajtu do paměti trvá **36 ns**

- Dokončení instrukce (nastavení příznaku přetečení, nuly, úklid po p
- Doplňkové akce spojené s dokončením instrukce trvají **19 ns**

K odměření času používá řadič hodinový signál

Každá fáze musí být provedena během jednoho taktu

Nejdelší zpoždění je 63 ns, takže perioda hodinového signálu (takt) musí trvat minimálně 63 ns

Navíc bývá zpravidla zvolena určitá rezerva a takt v našem případě by mohl trvat např. 100 ns

Procesor by tedy běžel na frekvenci 10 MHz

Frekvence hodinového signálu



- Jednotlivé fáze, na které je rozděleno provádění instrukce, obvykle netrvají stejně dlouho
- Ukažme si příklad na instrukci INC 100
- Tato instrukce má inkrementovat hodnotu bajtu, který leží v paměti na adrese 100
- Provádění této instrukce bude rozděleno do několika fází:
 - Přechtení strojového kódu instrukce z paměti (mikroprocesor musí zjistit, jakou instrukci má vlastně provádět)
 - Řadič generuje řídicí signál pro paměť, kterým vybírá adresu, ze které je potřeba přečíst strojový kód a dalším řídicím signálem sděluje, že z paměti se bude číst
 - Potřebný čas na přechtení strojového kódu z paměti: **25 ns**
 - Strojový kód je uvnitř mikroprocesoru dekodován a řadič plánuje provedení instrukce
 - Potřebný čas na dekodování přečteného strojového kódu: **12 ns**
 - Přechtení bajtu, který má být inkrementován, z paměti z adresy 100
 - Řadič generuje řídicí signál, kterým vybírá adresu 100 a dalším řídicím signálem sděluje, že z paměti se bude číst
 - Potřebný čas na přechtení datového bajtu z paměti: **25 ns**
 - Výpočet – Inkrementace
 - Řadič generuje řídicí signál, který říká ALU, aby inkrementovala zadanou hodnotu
 - ALU provádí výpočet (časově náročné)
 - Potřebný čas na provedení výpočtu: **63 ns**
 - Zápis výsledku zpět do paměti (inkrementovaný bajt se uloží zpět na adresu 100)
 - Řadič generuje řídicí signál, kterým vybírá adresu 100 a dalším řídicím signálem sděluje, že do paměti se bude zapisovat
 - Zápis bajtu do paměti trvá **36 ns**
 - Dokončení instrukce (nastavení příznaku přetečení, nuly, úklid po paměti)
 - Doplňkové akce spojené s dokončením instrukce trvají **19 ns**

Přestože některé fáze lze provést poměrně rychle (např. za 12 ns), následuje provádění další fáze vždy až v dalším taktu (po uplynutí 100 ns)

Správně navržený mikroprocesor má proto činnost rozloženou do pokud možno stejně dlouho trvajících fází

Výkon klasických mikroprocesorů



- Každý mikroprocesor potřebuje k vykonání každé instrukce několik taktů
- Některé instrukce bývají jednoduché a k jejich vykonání je potřeba méně taktů, jiné instrukce jsou složité a jejich provedení trvá procesoru více taktů
- Příklad:
- U **klasických mikroprocesorů** to vypadá například takto
 - Mikroprocesor běží na frekvenci **12 MHz**
 - Jednoduché instrukce (např. inkrementace) procesor vykoná za **12 taktů**, složitější instrukce (např. dělení dvou čísel) dokončí za **24 taktů**
 - Mikroprocesor tedy dokáže vykonat **1 milion jednoduchých** nebo **500 000 složitých** instrukcí za sekundu
 - Při zpracování konkrétního programu záleží na tom, kolik obsahuje jednoduchých a kolik složitých instrukcí, abychom mohli určit, kolik instrukcí za sekundu se vykoná v průměru (např. 725 000 Instr/s)
 - Počet vykonaných instrukcí za sekundu je podstatně nižší než taktovací frekvence



Příklad

- Mikroprocesor má taktovací frekvenci **100 MHz**
 - Vykonání instrukce trvá v průměru **10 taktů**
 - Jak dlouho trvá v průměru vykonání jedné instrukce?
 - Kolik instrukcí vykoná průměrně procesor za sekundu?
 - V kolikátém taktu po spuštění dokončí procesor desátou instrukcí?
-
- Instrukce jsou časově různě náročné. Provedení jedné instrukce vyžaduje v průměru 10 taktů. Vykonání instrukce tedy průměrně trvá $t_{instrukce} = 10 \text{ takt} = 10T = 10 \frac{1}{f} = 10 \frac{1}{100\,000\,000} = \mathbf{100 \text{ ns}}$
 - Instrukce se provádějí postupně. Provedení instrukce vyžaduje v průměru 10 taktů. Oscilátor během jedné sekundy generuje 100 000 000 taktů.
 - $IPS = \frac{f}{10} = \frac{100\,000\,000}{10} = \mathbf{10\,000\,000 \text{ Instrukcí za sekundu}}$
-
- Po spuštění procesoru trvá 10 taktů, než bude vykonána první instrukce. Provádění druhé instrukce bude dokončeno za dalších 10 taktů, tedy ve 20. taktu. Třetí instrukce bude hotová v 30. taktu atd... Desátá instrukce bude tedy dokončena ve **100. taktu**
-
- **Situace se ale může každou sekundu měnit, podle toho jak složité instrukce mikroprocesor právě vykonává – provedení každé instrukce procesoru trvá jinak dlouho**



- IPS = Instructions per second (počet instrukcí vykonaných za sekundu)
- MIPS = Million instructions per second
- Jedná se o primitivní ukazatel „výpočetního výkonu“
- Dva různé mikroprocesory se stejným IPS (stejným počtem instrukcí vykonaných za sekundu) vůbec nemusí mít stejný výpočetní výkon (například pokud jeden z nich je 64-bitový a druhý 8-bitový - pak i přes stejný počet vykonaných instrukcí bude rozdíl ve výkonu značný)
- Dokonce ani dva různé 64-bitové mikroprocesory se stejným IPS nebudou mít stejný výpočetní výkon (například pokud jeden z nich umí vykonávat pouze jednoduché instrukce a druhý je složitější a nabízí složitější operace)



Příklad

- Mikroprocesor má taktovací frekvenci 100 MHz.
- Vykonání jedné instrukce vyžaduje 5 taktů.
- *Určete výpočetní výkon mikroprocesoru v MIPS*
- Během jedné sekundy máme 100 000 000 taktů
- Každý pátý takt dokončí mikroprocesor jednu instrukci
- Počet vykonaných instrukcí za sekundu $IPS = \frac{100\,000\,000\, \text{taktů/s}}{5\, \text{taktů}} = 20\,000\,000$
- Mikroprocesor má tedy „výpočetní výkon“ 20 MIPS



Příklad

- Mikroprocesor má výpočetní výkon 1 MIPS
- Vykonání každé instrukce vyžaduje 6 taktů
- *Určete taktovací frekvenci mikroprocesoru*
- *Určete dobu provádění jedné instrukce*
- *Určete dobu trvání jednoho taktu*
- Mikroprocesor vykoná 1 000 000 instrukcí za sekundu
- Vykonání každé instrukce trvá 6 taktů, takže vykonání milionu instrukcí vyžaduje 6000000 taktů
- Aby se to stihlo za sekundu, musí proběhnout 6 000 000 period hodinového signálu
- Taktovací frekvence je tedy 6 000 000 Hz = 6 MHz
- Jeden takt trvá $T = \frac{1}{f} = \frac{1}{6\,000\,000} = 166,6\text{ ns}$
- Vykonání jedné instrukce trvá 6 taktů, $t_{instrukce} = 6T = 6 \cdot 166,6\text{ ns} = 1000\text{ ns} = 1\mu\text{s}$
- Anebo jinak -za sekundu se vykoná 1000000 instrukcí, takže $t_{instrukce} = \frac{1\text{ s}}{1\,000\,000} = 1\mu\text{s}$



Příklad

- Mikroprocesor během jedné sekundy vykonal 1 000 000 jednoduchých a 1 500 000 složitých instrukcí
 - Provedení jednoduché instrukce vyžaduje 5 taktů
 - Provedení složité instrukce vyžaduje 10 taktů
 - *Určete taktovací frekvenci mikroprocesoru*
-
- Mikroprocesor vykonal 1 000 000 jednoduchých instrukcí, což vyžadovalo $1\,000\,000 \times 5 \text{ taktů} = 5\,000\,000 \text{ taktů}$
 - Mikroprocesor vykonal 1 500 000 složitých instrukcí, což vyžadovalo $1\,500\,000 \times 10 \text{ taktů} = 15\,000\,000 \text{ taktů}$
 - Vykonání všech instrukcí tedy vyžadovalo $5\,000\,000 + 15\,000\,000 \text{ taktů} = 20\,000\,000 \text{ taktů}$
 - Toto se stihlo během jedné sekundy
 - Taktovací frekvence tedy musí být 20 000 000 taktů za sekundu
 - $f = \underline{20 \text{ MHz}}$

Výkon moderních mikroprocesorů



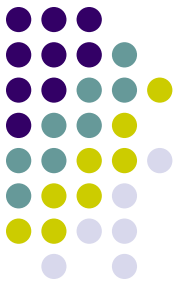
- U dnešních **moderních mikroprocesorů** je situace odlišná a mnohem komplikovanější
- Nadále platí, že mikroprocesor potřebuje k vykonání každé instrukce **několik taktů**
- Protože ale moderní mikroprocesory zpracovávají **několik instrukcí současně**, je možné, že v každém taktu je dokončena jedna nebo i více instrukcí
- U **moderních mikroprocesorů** se tedy setkáme s tím, že dokáží provést více instrukcí za sekundu, než jaká je jejich taktovací frekvence
- Např. mikroprocesor běží na frekvenci **2,5 GHz** (2,5 miliardy taktů za sekundu) a při tom dokáže vykonat v průměru **4 miliardy** instrukcí za sekundu

Taktovací frekvence



- **Nominální frekvence** mikroprocesoru je frekvence, na které výrobce zaručuje bezproblémový chod
- Snížením této frekvence dojde k „**podtaktování**“, což má za důsledek **nižší výpočetní výkon**, než kterého je mikroprocesor schopen
- Podtaktováním se také často významně sníží **elektrický příkon** procesoru a je tedy **úspornější**
- Moderní mikroprocesory při nízké zátěži (když počítač zahálí) umí automaticky přechodem na nižší taktovací frekvenci přejít do **úsporného režimu běhu**
- Zvýšením taktovací frekvence nad doporučenou hodnotu dochází k „**přetaktování**“
- Přetaktováním lze **zvýšit výpočetní výkon mikroprocesoru**, ale bude docházet k větší spotřebě elektrické energie a **zahřívání**
- Dokážeme-li mikroprocesor uchládit, lze obvykle taktovací frekvenci o dost zvýšit, protože rezerva zvolená při stanovení nominální frekvence bývá značná
- Nad určitou hraniční taktovací frekvencí však již procesor přestane zcela fungovat, i kdybychom ho chladili sebevíc
- Problém není v zahřívání, ale v tom, že procesor „přestane stíhat“ – Frekvence je moc vysoká, perioda hodinového signálu je moc krátká a jednotlivé fáze provádění instrukcí nastávají dříve než byly dokončeny předchozí, takže procesor generuje chaotické výsledky
- **Mezní frekvence** je taková frekvence, kdy **perioda hodinového signálu** je kratší, než **zpoždění** vznikající uvnitř mikroprocesoru – řadič generuje řídicí signály rychleji, než jsou na ně jednotlivé funkční bloky schopny reagovat

Frekvence



- Taktovací frekvence prvních mikroprocesorů se pohybovala okolo **500 kHz**
- Nejvyšší taktovací frekvence moderních mikroprocesorů se pohybuje okolo **4 GHz**
- Procesory běžící na **vyšší frekvenci** je velmi obtížné **ochlazovat**
- Moderní mikroprocesory by stíhaly běžet až na frekvencích 10 GHz, ale prakticky to není možné, protože obsahují miliardy tranzistorů, které musí být umístěny na velmi malé ploše, aby mezi nimi stíhal putovat signál (šířící se konečnou rychlostí)
- Při frekvenci 10 GHz stihne světlo během jednoho taktu urazit vzdálenost pouze 3 cm – vzdálenost dvou bodů, které si chtějí vyměnit informaci během jednoho taktu nemůže být větší
- Elektrický signál uvnitř mikroprocesoru se ale šíří nižší rychlostí, než je rychlost světla a navíc cesty jsou pořádně komplikované
- Na velmi malé ploše tak vzniká ohromné množství tepla, které nedokážeme odvádět a mikroprocesor se přehřívá
- Za posledních 15 let frekvence procesorů téměř nestoupá a zřejmě dále stoupat nebude

Základní parametry mikroprocesorů

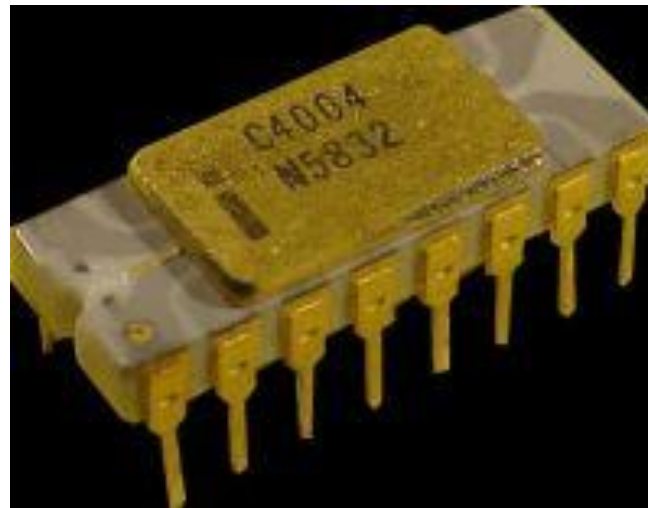


- Počet bitů (šířka zpracovávaných dat, registrů a ALU)
- Počet instrukcí v instrukční sadě
- Počet registrů
- Role registrů (střadačová architektura / univerzální registry)
- Možnosti adresace paměti (šířka adresy v bitech)
- Počet tranzistorů na jednom čipu
- Pouzdro a počet vývodů
- Napájecí napětí a příkon
- Taktovací frekvence (počet hodinových cyklů za sekundu – pozor, neodpovídá počtu provedených instrukcí, protože různé instrukce vyžadují různý počet taktů)
- Frekvence vnější sběrnice (komunikace s vnějším okolím procesoru může probíhat na jiné frekvenci /obvykle nižší/ než je taktovací)
- Průměrný počet vykonaných operací za sekundu
- Struktura procesoru (mikroprogram nebo obvod, FPU, RISC nebo CISC, cache – všechny tyto pojmy probereme později)

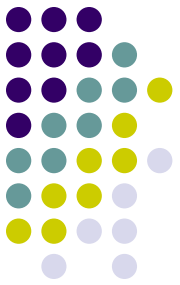


Počátky mikroprocesorů

- První obvod nazvaný **mikroprocesor** uvedla na trh firma Intel v roce 1970. Šlo o 4bitový procesor Intel 4004
- Souběžně s ním byl firmou Zilog vyvíjen systém MCS4
- Autoři obou procesorů pracovali střídavě pro obě firmy



Intel 4004



- Mikroprocesor byl dodáván v pouzdře Cerdip se 16 vývody
- Obsahoval 2 300 tranzistorů
- Napájecí napětí 15 V
- Maximální taktovací frekvence 750 kHz
- Instrukční sada obsahuje 45 instrukcí
- Sada registrů obsahuje 16 registrů s šířkou 4 bity (tzn. pracuje pouze s hodnotami 0..15 !)



8-bitové procesory



- V roce 1972 byl MCS8 od Zilogu prvním 8bitovým mikroprocesorem
- V roce 1972 také došlo k rozšíření datové sběrnice 4004 na 8 bitů - vznikl procesor 8008
 - Pouzdro DIP 18
 - 3500 tranzistorů
 - šestnáct 8-bitových registrů
 - Taktovací frekvence 500 až 800 kHz
 - Provádí 45000 až 100000 instrukcí za sekundu (různé instrukce jsou dle složitosti prováděny během různého počtu hodinových taktů)
- Procesor vykonal za sekundu zhruba stejný počet instrukcí jako jeho předchůdce i4004, přesto byl asi 3x výkonnější
- Vyšší výkonnost tedy není dána pouze počtem vykonávaných instrukcí, ale je třeba také posoudit o jak kvalitní instrukce jde a s kolikabitovými daty jsou prováděny
- 8008 zpracovává celý bajt, zatímco 4004 pracovala pouze se čtyřmi bity
- Krom toho jsou instrukce procesoru 8008 výkonnější (vyžadují méně hodinových taktů)

i8080



- V roce 1974 uvádí Intel svůj velmi úspěšný procesor 8080
- Stal se základem prvních osmibitových počítačů, stal se nejrozšířenějším procesorem sedmdesátých let a byl kopírován mnoha dalšími výrobci (např. i československou Teslou)
 - 4500 tranzistorů
 - Osmibitové registry: A (střadač), B, C, D, E, H, L
 - Taktovací frekvence 1 až 2 MHz
 - provádí cca 250000 operací za sekundu



Motorola 6800



- první z rodiny 6800, která je v podstatě až dodnes základem počítačů Apple Macintosh
- Dnes z něj vycházejí řady jednočipů 68HC11 a 68HC08
 - 4000 tranzistorů
 - Instrukční sada 75 instrukcí
 - Pouzdro DIL 40
 - 8-bitová data, 16-bitové adresy
 - Taktovací frekvence až 2 MHz



MOS 6502



- Legendární čip, který se proslavil zejména svou **cenou**
- V době, kdy Intel prodával 8080 za 179 dolarů a Motorola svoji 6800 také za 179 dolarů, přišel MOS s cenou pouhých 25 dolarů
- MOS 6502 či jeho upravené verze MOS 65C02 či MOS 6510 byly použity v mnoha osobních i domácích počítačích
 - především v Apple II (jeden z nejprodávanějších počítačů v USA)
 - Apple Lisa (první osobní počítač se grafickým uživatelským rozhraním)
 - osmibitových Atari
 - Commodore C64





Commodore 64 – klasika 80. let

Počítač pro domácí použití byl k sehnání i v
Československu. Ve světě se prodalo 25 milionů kusů



AMD Am2900



- První „stavebnice“ mikroprocesoru firmy **AMD** z roku 1975
- Technologie **Bit Slicing** - „krájení bitů“
- Konstrukce procesoru z modulů schopných pracovat s menší bitovou šířkou, než je výsledný počet bitů procesoru
- Obvody řady 2900 schopny tvořit řídicí jednotky s daty, adresami a instrukcemi, jejichž bitová šířka mohla být jakýkoli násobek 4 bitů
- Každý z těchto modulů zpracovává jednu část bitového slova
- Tyto moduly schopny zpracovat „plné“ bitové slovo (dané šířkou datové sběrnice) nebo obecně libovolně dlouhé bitové slovo
- Procesory založené na této technice se většinou skládají z 1, 2, 4 nebo 8bitové aritmeticko-logické jednotky (ALU) – v případě AMD 2900 jde o 4 bity
- Ale teoreticky nebyl problém postavit si ze „stavebnice“ obvodů 64 bitový procesor nebo i více...
- Jedním hlavním problémem s touto modulární technikou byl větší požadovaný počet obvodů, který by jinak zastal jeden procesor.
- Postupem času a rozvojem technologie přestal být tak velký problém vyrábět procesory se všemi integrovanými obvody ve společném jádru

Z80



- Uveden firmou **Zilog** v roce 1976
- Struktura registrů i instrukční sada vycházely z architektury procesoru Intel8080
 - 8500 tranzistorů na čipu
 - 8-bitová data, 16-bitů adresa (tzn. umí adresovat paměť až 64 kB)
 - Taktovací frekvence až 4 MHz (vylepšený model Z80A)
 - Instrukční soubor obsahuje 158 instrukcí
 - Pouzdro DIL se 40 vývody
 - Napájecí napětí +5 V
- Byl použit v nejprodávanějším osmibitovém počítači Sinclair ZX Spectrum
- V Československu byl Z80 použit v domácím mikropočítači Didaktik (cena v roce 1988 asi 6000 Kčs)

ZX Spectrum



Texas Instruments TM9900



- První **16-bitový** procesor vyrobila firma Texas Instruments v roce 1976
- Předběhlo ji vlastně trochu AMD se svou stavebnicí Am2900, ze které se dala postavit 16-bitová konfigurace, ale vyžadovalo to kaskádové propojení několika obvodů, zatímco TM9900 byl jediný čip
- Umí zpracovávat 16-bitová data, ale pracuje pouze s 15-bitovými adresami (tzn. max. 32 kB paměti)





Přelom 70. a 80. let dvacátého století

- Někdy v této době se oddělila vývojová větev monolitických mikroprocesorů (jenočipových mikropočítačů)
- 1980 vzniká jednočip i8051



Kontrolní otázky

- Vysvětlete rozdíl mezi 8-bitovým a 16-bitovým mikroprocesorem
- Kdy vznikají první mikroprocesory ?
- Kolik tranzistorů obsahují první mikroprocesory ? Kolik tranzistorů současné ?
- Co je to programový čítač ?
- K čemu slouží příznak přetečení ?
- Vysvětlete rozdíl mezi střadačovou architekturou a univerzálními registry
- Jak velkou paměť lze připojit k osmibitovému mikroprocesoru s patnáctibitovou adresací ?
- Mikroprocesor má registry s šířkou 16 bitů a umí adresovat 1 MB paměti. O kolikabitový mikroprocesor se jedná ? Kolikabitovou adresaci tento mikroprocesor používá ?
- Vysvětlete rozdíl mezi uložením dat v registrech a v paměti
- K čemu slouží na většině mikroprocesorů registr SP ?
- Jaké firmy spolupracovali na výrobě prvního mikroprocesoru ? Pod jakými názvy se tento mikroprocesor vyráběl ?
- Vyjmenujte alespoň tři legendární osmibitové mikroprocesory
- Dva různé mikroprocesory A a B vykonají 20 milionů instrukcí za sekundu. Mikroprocesor A má taktovací frekvenci 100 MHz. Mikroprocesor B má taktovací frekvenci 200 MHz. Který z těchto mikroprocesorů má vyšší výpočetní výkon.



Kontrolní otázky

- Kolikrát vzrostla frekvence během vývoje mikroprocesorů od prvního procesoru i4004 po dnešní nejmodernější ?
- Mikroprocesor i8051 běží na frekvenci 6 MHz. Instrukce násobení trvá 48 taktů. Kolik násobení je procesor schopen vykonat na sekundu ?
- Jak se nazývá obvod, který generuje hodinový signál ?
- Mikroprocesor je synchronní/asynchronní sekvenční/kombinační logický obvod ?
- Co se obvykle stane, pokud taktovací frekvenci mikroprocesoru zvýšíte o 10 % ?
- Co se obvykle stane, pokud taktovací frekvenci mikroprocesoru zvýšíte na dvojnásobek ?
- Co se obvykle stane, pokud taktovací frekvenci mikroprocesoru snížíte na polovinu ?
- Zjistěte, na jaké frekvenci běžel mikroprocesor Pentium IV v roce 2003
- Zjistěte, na jaké frekvenci běží současné nejvýkonnější běžně používané čtyřjádrové mikroprocesory
- Co je to bit-slicing ?
- Co to znamená DIL 40 ?
- Seřadte tyto mikroprocesory dle roku jejich uvedení: Z80, TM9900, Intel 4004, Intel 8080, Intel 8008