

Okruh 3 otázka 7

Distribuované řídicí systémy, architektura paralelních počítačů na bázi transputerů. Programovací jazyk OCCAM. Asynchronní komunikace, deadlock. Topologie distribuovaných systémů. Výpočet časové náročnosti paralelních algoritmů. Globální komunikace. Průmyslové distribuované systémy. Průmyslové sítě. Profibus., CAN, CAL, Device Net, P-Net, LonWorks,. Modelování distribuovaných řídicích systémů pomocí Petriho sítí.

Distribuované řídicí systémy

- nasazování mikropočítačů do řízení technologických celků – propojení automatizovaných systémů
- důležitým prvkem je komunikace, díky komunikaci mezi jednotlivými technologickými částmi lze mluvit o distribuovaném systému
- snažíme se komunikaci co nejvíce omezit - využitím „lokální inteligence“ to lze (zvýšení lokálního rozhodování a zpracování dat; datová komprese; efektivní komunikační protokoly)

Vlastnosti

Obecně

Všechny funkce zpracování a uložení informací, řízení I/O systémů atd. jsou rozptýleny mezi jednotlivé stanice zpracování dat. Distribuované systémy obsahují větší počet takovýchto zpracovatelských prvků. Prvky jsou v geografickém nebo funkčním uspořádání. Jsou propojeny počítačovou sítí. Vzájemně mezi sebou komunikují za účelem dalšího zpracování dat a vykonávání řízení podle požadavků uživatele.

Nevýhody

Potřeba složitější komunikace mezi jednotlivými prvky, problematika integrity a ochrany dat, heterogenita použitých technických a programových prostředků.

Výhody

Pružnost a rozšiřitelnost, otevřenost

Vzhledem k modularitě lze systém jednoduše měnit a rozšiřovat. Možnost sdílení zdrojů a periférií.

Modularita

Každý prvek poskytuje dobře definované rozhraní.

Spolehlivost

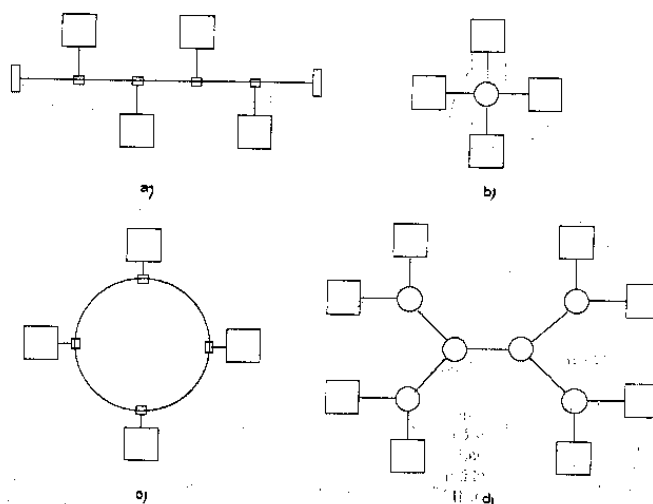
Potenciální schopnost pokračovat při výpadku některé části = vyšší provozuschopnost

Výkonnost

Zkrácení doby odezvy ve srovnání s centralizovaným řízením, multiprocesorové systémy.

Topologie

Základní topologie:



- sběrnice* – vhodné pro lokální sítě, vyšší stupně měřících a řídicích systémů, nevýhodné jako průmyslová sběrnice – není zaručen čas pro přístup ke sběrnici
- hvězdicová* – při výpadku jedné stanice ostatní bez problémů komunikují, při výpadku uzlu - problém
- kruhová* – výhodné pro optická média (každá stanice má jeden vstup a jeden výstup), při výpadku jakékoli stanice nastává problém. Zaručena maximální horní hranice čekání na sběrnici.
- Stromová* - komplikovanější rozšíření sběrnice struktury. Nutno definovat max. zpoždění mezi dvěma nejvzdálenějšími stanicemi v síti.

Architektura paralelních počítačů na bázi transputerů

Paralelní počítače

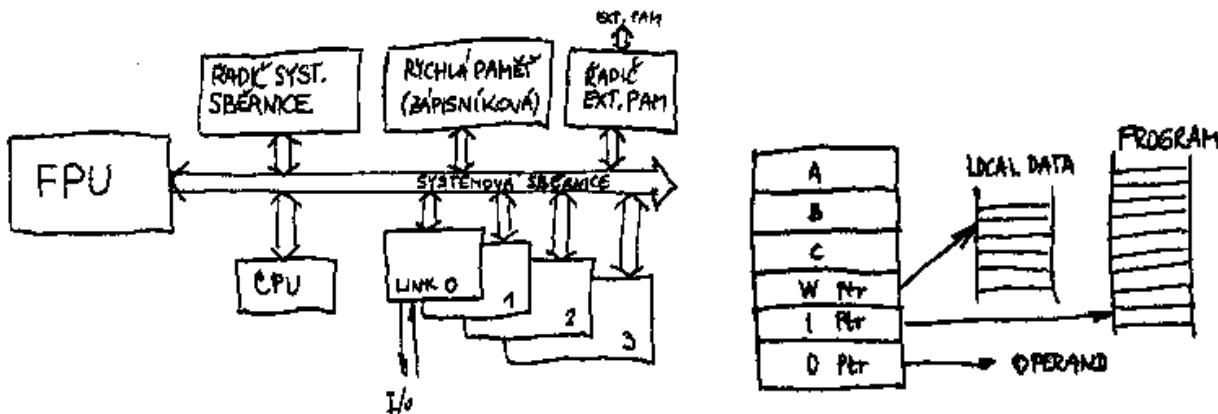
Použití

Meteorologie, zpracování obrazu, obecně analýza rozsáhlého množství dat

Nevýhody

Morální zastarávání – než se vyvine program pro paralelní počítač, sekvenční počítače se mnohonásobně zrychlí.

Architektura transputerů



Transputer tvoří FPU (Floating Point Unit), jednoduchá CPU, rychlá interní zápisníková paměť, řadič externí paměti, rychlá systémová sběrnice, řadič systémové sběrnice, pro komunikaci s ostatními transputery a okolím slouží rychlé komunikační kanály – LINK1, LINK2...

CPU – tři registry A,B,C organizované do zásobníku

FPU – tři registry AF,BF,CF

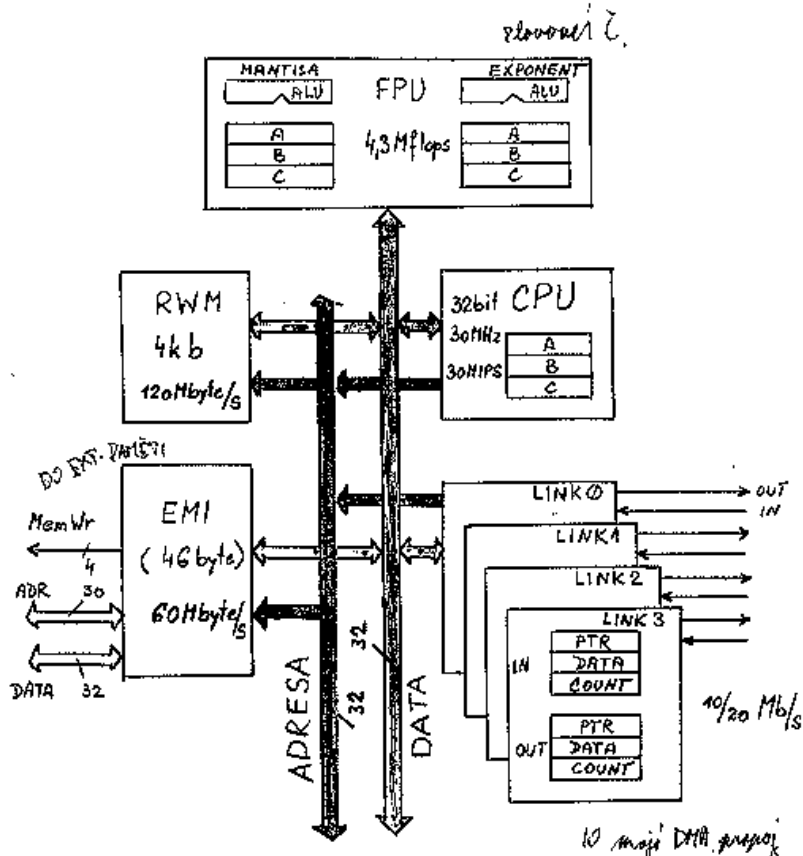
WPtr – Workspace Pointer – ukazuje na aktuální lokální data (v zápisníkové paměti)

IPtr – Instruction Pointer – ukazuje na instrukci programu

OPtr – Operand Pointer nebo Operand Register – příprava operandu instrukce

Charakteristika transputerů:

- rychlá vnitřní paměť, malé množství registrů => jednoduchá řídicí logika
- výpočty se provádějí na vrcholu zásobníku (není zapotřebí adresovat)
- malé množství instrukcí (RISC) **Příklad konkrétního transputeru – INMOS T801:**



Programovací jazyk OCCAM

- používá se např. v transputerech řady T800, T9000
- umožňuje psát paralelní algoritmy a jejich implementaci na síť procesorů
- aplikace je napsána jako skupina procesů, které běží současně a komunikují spolu prostřednictvím kanálů

Možnosti zpracování programu

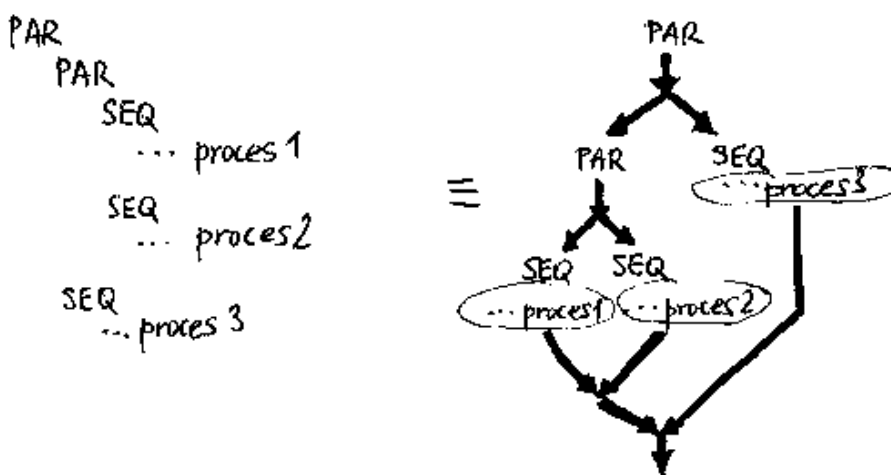
- SEQ – příkazy se vykonávají sekvenčně
- PAR – program probíhá paralelně ve větvích. Chod všech větví je na konci synchronizován => program pokračuje až po ukončení všech větví, nelze dynamicky zakládat procesy, nelze použít rekurzi.
- ALT – vybírá připravenou větev a provádí odpovídající akci. Když není žádná větev připravena – čeká

Datové typy

- BOOL, BYTE, INT, INT16, INT32, REAL 32, REAL 64

Příklad definice procesů v OCCAMu

DEFINICE PROCESŮ V OCCAMU :



Asynchronní komunikace

Komunikace mezi procesy řízená událostmi, signály. Nedá se vždy analyzovat možný počet stavů. Při velkém množství komunikačních kanálů a složitosti řídicího systému – nebezpečí deadlocku.

Deadlock

Totální zablokování systému, stav havárie, ze kterého se systém není schopen dostat bez vnější pomoci (např. RESET) Nejčastěji deadlock způsobí nechtěná nekonečná smyčka (čekání na kombinaci, která již nepříjde apod...).

Jak předejít deadlocku:

Lze analyzovat všechny stavy systému (velká časová náročnost – čas na analýzu roste exponenciálně se složitostí systému). V praxi – při návrhu programů se podřídí určité struktuře – omezení oboustranné komunikace (klient/server), použití timeoutů apod.

Výpočet časové náročnosti paralelních algoritmů

n – počet procesorů

w – rozsah řešeného problému

$T_s(w)$ – sekvenční výpočetní čas

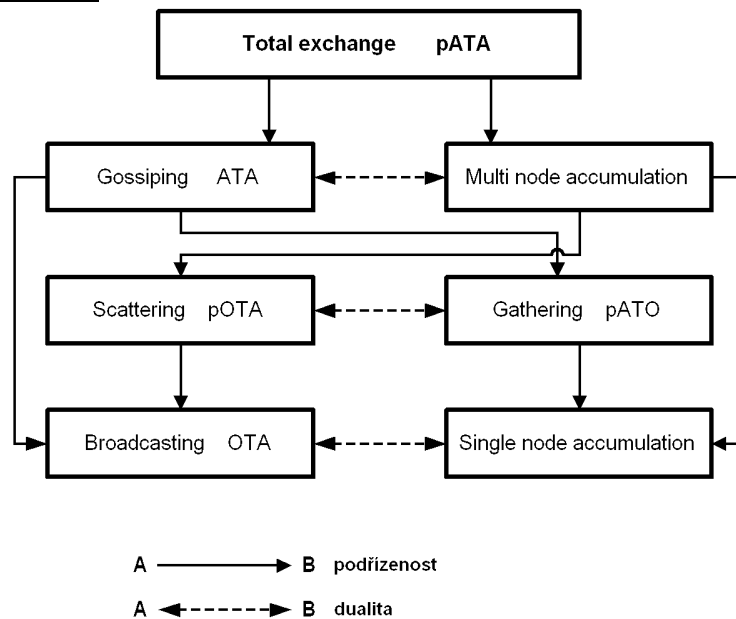
$T_p(w, n)$ – výpočetní čas paralelního algoritmu

$$\text{Speedup} \quad ; \quad S(w, n) = \frac{T_s(w)}{T_p(w, n)} \quad \text{Efektivita} \quad E(w, n) = \frac{S(w, n)}{n}$$

$$\text{Amdahlsovo pravidlo: } ; \quad S(n) \leq \lim_{n \rightarrow \infty} \frac{1}{f + \frac{1-f}{n}} = \frac{1}{f} \quad f \dots \text{sekvenční část algoritmu}$$

Při posuzování časové náročnosti algoritmů se bere v úvahu: doba výpočtu, doba komunikace, synchronizace a prodlevy jednotlivých dějů. Výkonnost paralelních procesů obecně závisí na poměru *výpočet / komunikace*.

Globální komunikace



podřízenost: algoritmus řešící A může řešit i B
dualita: problém A může být řešen algoritmem B s obráceným časem
ATO – all to one, **OTA** – one to all, **pATO**, **pOTA** – personalized ...

Broadcasting

OTA, difusion, tentýž paket pro všechny nody

Gossiping

ATA, „drby“, multinode broadcasting, všechny nody provádějí naráz broadcasting, všichni musí vyměnit informaci se všemi, aby byl přenos informace úplný

Scattering

pOTA, jeden nód vysílá specializované pakety pro všechny ostatní nody - každému nódu zvláštní paket

Total exchange

pATA, „vánoce – dárek každý každému“, všechny nody provádějí scattering

Gathering

pATO, do jednoho nódu směřují pakety ze všech ostatních nódů

Single node accumulation

Všechny nody pošlou pakety do jednoho nódu s tím, že pakety mohou být kombinovány (např. sčítání hodnot), takže čas pro komunikaci kombinovaného paketu je stejný, jako pro komunikaci kteréhokoli z původních paketů.

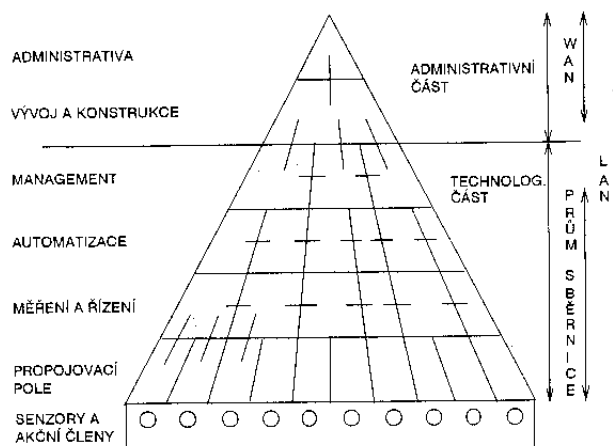
Multi node accumulation

Single node accumulation současně na všech nodech

Nejčastější způsob komunikace mezi nody, které nemají přímé spojení:

Zpráva se dočasně uloží v mezinódu a pak pošle dál, než doputuje do cílového nódu – tzv. „Store and forward“.

Průmyslové distribuované systémy



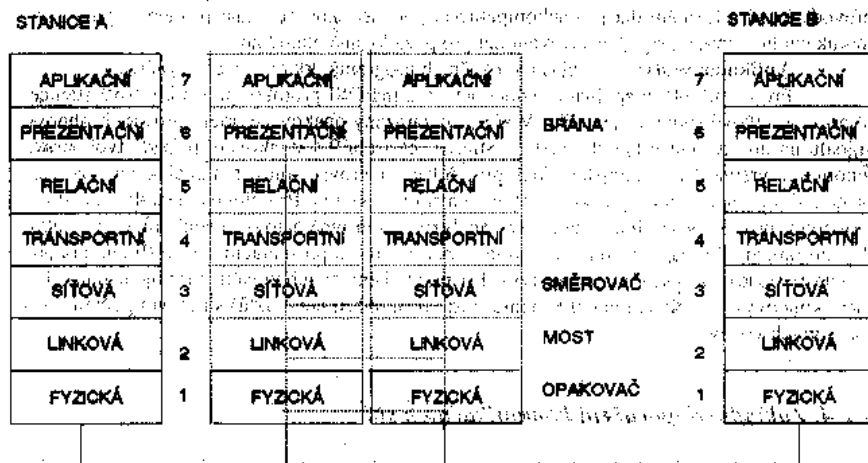
Průmyslové sítě

ISO/OSI (International Standardization Organization / Open System Interconnection)

- členění do vrstev, které mezi sebou komunikují pomocí služeb

Sedm vrstev ISO/OSI

- **Fyzická vrstva** – mechanické a elektrické vlastnosti přenosového media, přenosové medium, kódování modulace
- **Linková vrstva** – přístup na sběrnici, stavba paketů (rámců), kontrola přenosu, opakování při chybě
- **Síťová vrstva** – propojení mezi sítěmi
- **Transportní vrstva** – navazování spojení, rozklad zpráv do paketů
- **Relační vrstva** – logické spojení mezi účastníky
- **Prezentační vrstva** – převody kódů, komprese, dekomprese
- **Aplikační vrstva** – návaznost na aplikaci



Detekce chyb

- kontrolní součet
- cyklický kód (Cyclic Redundant Check)
- samoopravné kódy

Formát přenosového protokolu

- bitově orientované protokoly – start, data, stop
- znakově orientovaný protokol – data, parita, stop bit
- blokově orientované protokoly – data organizována do bloků, řídicí znaky

Dělení sítí podle přístupu na sběrnici

- centralizovaný – master-slave
- decentralizovaný
 - deterministický – zařízení si vyměňují pověření k vysílání (token) (tokenbus, token ring)
 - stochastické (náhodný přístup) – CSMA/CD (Carrier Sense Multiple Access / Collision Detection)

Profibus

dva protokoly

- řízení procesů na stejné úrovni
- řízení nadřazeného zařízení

požadavky na síť

- odolnost proti rušení
- schopnost se automaticky zastavit při výpadku
- možnost rozšiřování bez přerušení
- otevřenost
- počet účastníků 0 – 126 (bez opakovačů)
- Rychlost – 9,6 kB/s – 12MB/s
- Přenos – asynchronní poloduplex
- Maximální délka zprávy 0 – 255 bytů
- token bus
- spojení – RS485, optika

Profibus FMS – přenos dat mezi nekompatibilními systémy

Profibus DP – pro nekvalifikované uživatele

CAN – Controller Area Network

- sériový komunikační protokol
- sběrnice s náhodným přístupem a s bezkolizním řešením konfliktů na základě prioritního rozhodování
- přenosová rychlost do 1MBitu/s
- délka datového pole 8 bitů

Typy zařízení

Aktivní - vysílá, přijímá, může přerušit vysílání jiných

Pasivní - vysílá, přijímá, nemůže přerušit vysílání jiných

Odpojené - nemůže nic

Použití

automatizace výroby

procesní řízení

automobilový průmysl

Detekce chyb

chyba úrovně signálu – zpětné čtení

nepotvrzení příjmu zprávy

CAL

- objektově orientovaný přístup
- komunikace typu client – server pro výměnu aplikačních dat
- komunikace typu master – slave pro konfiguraci systému

CAL NMT

služby pro správu sítě zařízení

komunikační model master – slave

povolený jediný NMT správce sítě – NMT Master

Device Net

Jedná se o poměrně levný způsob připojení průmyslových zařízení jako jsou spínače, fotoelektrické buňky, elektromotory, startéry, displeje k PLC a PC. Síť je navržena tak, aby propojení jednotlivých prvků bylo co nejjednodušší. Podrobnosti - kvalitní PDF materiál na http://216.10.36.18/10_2/05_tech/DNET-Overview.pdf (cca 250 kB)

Přenosové rychlosti

125 kbps/500 m 250 kbps/250 m 500 kbps/100 m

ISO/OSI

Device NET je 7.-aplikační vrstvou podle ISO/OSI. Je nadstavbou nad protokolem CAN-BUS, který zde tvoří linkovou vrstvu.

P-Net

- sběrnice typu multimaster
- jednoduché spojování sběrnic (zachovává ISO/OSI)
- podpora vyššího programovacího jazyka
- inteligentní podřízené moduly – do modulu lze nahrát program

Použití – mlékárenský průmysl; výroba domácích spotřebičů (BOSCH, Siemens)

Fyzická vrstva – RS485; fyzický kruh bez zakončení; stíněná dvoulinka; na sběrnici max. 125 zařízení; kódování NRZ

Linková vrstva – vytváří rámce; asynchronní přenos; max. 32 masterů; předávání tokenu

Síťová vrstva – zajišťuje spojení více sběrnic bez opakovačů

Transportní vrstva – kódování příznaků

Aplikační vrstva – zprostředkovává přístup k proměnným - program

LonWorks

Použití - automatizace budov (inteligentní dům), metropolitní síť

Protokol LonTalks

- má 2 až 7 vrstvy ISO/OSI. (nemá standardní fyzickou vrstvu – je libovolná)
- Fyzická vrstva – může být libovolná (nemá ji standardně zadanou) (RS485; infračervené)
- Linková vrstva – synchronizace, modulace, volitelná priorita, p-naléhající
- Síťová vrstva – adresace (neuron – jeden modul; doména – podsíť); topologie libovolná
- Transportní vrstva – nepotvrzované spojení, číslování paketů

Modelování distribuovaných řídicích systémů pomocí Petriho sítí

Petriho sítě – grafický a matematický nástroj pro modelování DES (Discrete Event System – systémy diskretních událostí)

• Grafy sestávají ze tří typů prvků



- **Uzly** - umožňují popsat stavy systému
- **Přechody** - modelují možnost změny stavu systému
- **Orientované hrany** - propojují uzly a přechody, definují cesty

• Uzly a přechody se v průběhu časy střídají

• V čase t každý uzal P může obsahovat celé číslo $N(P)$ značek

• **Značkování N sítě** je určeno **značkovacím vektorem**

$$N = (N(P_1) \dots N(P_n))$$

kde $P_1 \dots P_n$ jsou uzly sítě představující stavy

• **Počáteční značení N_0** popisuje počáteční stav systému

• **Vývoj systému** je reprezentován přesunem značek v síti na základě aktivace přechodů

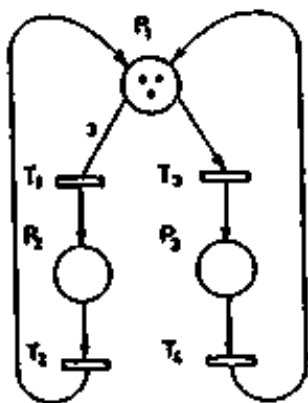
• Každé nové značení reprezentuje nový stav systému

Na rozdíl od GRAFCETu každý uzal obsahuje nejen kvalitativní informaci (aktivní/neaktivní) ale i kvantitativní informaci (kolikrát aktivní/neaktiv.)

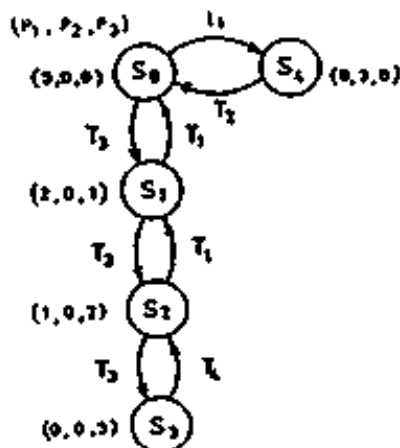
• Každý stav systému může být modelován Petriho sítí s několika uzly a jednou nebo více značkami

• Úzel nepředstavuje jeden stav systému, ale je částí jednoho nebo více stavů v průběhu jejich značení

• Množina všech dosažitelných značení a stavový diagram jsou ekvivalentní



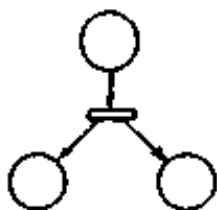
Petri síť



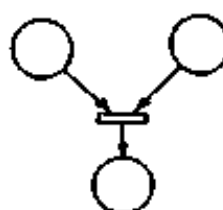
Stavový diagram

vidla a vlastnosti

Každý přechod má jednu nebo více vstupních a výstupních hran



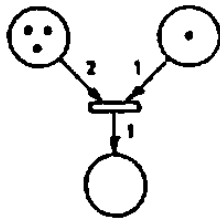
račítka **paralelního** vývoje sekvencí



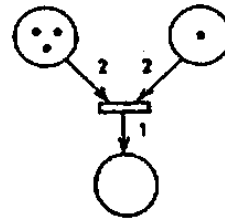
synchronizace paralelních sekvencí

Kladné číslo připojené k hraně udává váhu hrany. Mají-li všechny hrany váhu 1 - ordinární (symetrická) Petriho síť, při rozdílných - obecná Petriho síť

Přechod je uvolněn - každý jeho vstupní uzel má značení \geq váze hrany spojující uzel a přechod



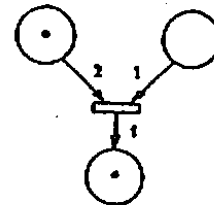
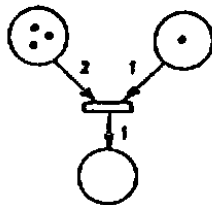
uvolněný přechod



neuvolněný přechod

Aktivace přechodu - sestává ze dvou kroků

- ze vstupních uzlů přechází do přechodu počet značek odpovídající váze hrany vstupu
- z přechodu přechází do výstupního uzlu počet značek odpovídající váze výstupní hrany



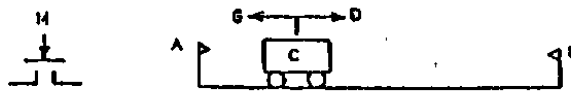
aktivace přechodu
(změna značení)

* změna značení dovoluje analyzovat dynamické chování modelového systému

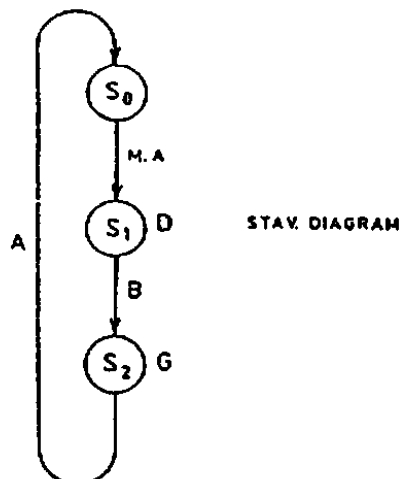
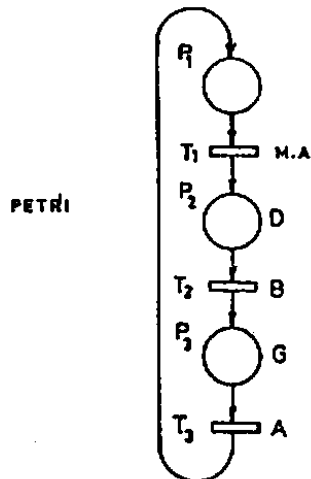
* z počátečního značení M_0 aktivací přechodů můžeme získat množinu dosažitelných značení M_0^* . Každé z nich je $M \in M_0^*$ a odpovídá jednomu stavu modelovaného systému

Příklady :

1. Pohyb vozíku C po dráze mezi body A a B, pohyb fidi G a D, start tlačítkem M



automatický návrat M: A \rightarrow B \rightarrow A



Vlastnosti a druhy Petriho sítí

PS jsou:

Bipartitní – 2 druhy uzlů (přechody a místa); každá hrana má jeden koncový bod uzel a druhý přechod (nelze spojit dvě místa nebo dva přechody).

Orientované – hrany jsou orientované.

Ohodnocený graf – hranám lze přiřadit čísla, která určují váhu.

Vývoj systému je reprezentován přesunem pešků po aktivaci přechodů.

Každé nové značení reprezentuje stav systému.

Výhody PS – větší modulárnost, menší velikost (neroste nekontrolovaně).

Neoznačená síť

- uspořádaná čtveřice $A = \langle P, T, Pre, Post \rangle$
- P – vektor míst
- T – vektor přechodů
- Pre – Matice $N \times M$ specifikuje hrany vycházející z míst do přechodů
- $Post$ – Matice $N \times M$ specifikuje hrany vycházející z přechodů do míst

Označená síť

- uspořádaná pětice $A = \langle P, T, Pre, Post, M_0 \rangle$
- M_0 – značka (pešek)

Aktivace přechodu

Přechod je uvolněn, jestliže ve všech jeho vstupních místech je počet tokenů větší nebo roven váze hrany jdoucí z místa do přechodu.

Aktivace přechodu odebere tokeny z vstupních míst přechodu a podle vah z přechodů vystupujících přidá tokeny do výstupních míst.

Autonomní síť (Petriho)

obecně popisuje kvalitativní postupy, nezávisí na okolí

Neautonomní síť

závisí na časech, na vstupech apod.

Ohraničená síť

Může být modelována konečným automatem.

Neohraničená síť

Nemůže být modelována konečným automatem.

Pseudoživá síť

Síť je pseudoživá, jestliže všechny její přechody jsou pseudoživé. Pseudoživý přechod – přechod, který může být od počátku značení přeskóčen alespoň jednou.

Živá síť

Síť je živá, jestliže všechny její přechody jsou živé. Živý přechod – jestliže pro všechna dosažitelná značení existuje taková sekvence, která umožňuje přeskok tohoto přechodu.

Reverzibilní Petriho síť

Síť je reverzibilní, jestliže pro všechna dosažitelná značení existuje taková sekvence, která převede síť do původního stavu

PS s inhibovanou hranou

Přechod je uvolněn, jestliže počet tokenů ve vstupních místech spojených s přechodem obyčejnou hranou je dostatečný a ve vstupních místech spojených s přechodem inhibovanou hranou je počet tokenů nulový

PS časované (neautonomní)

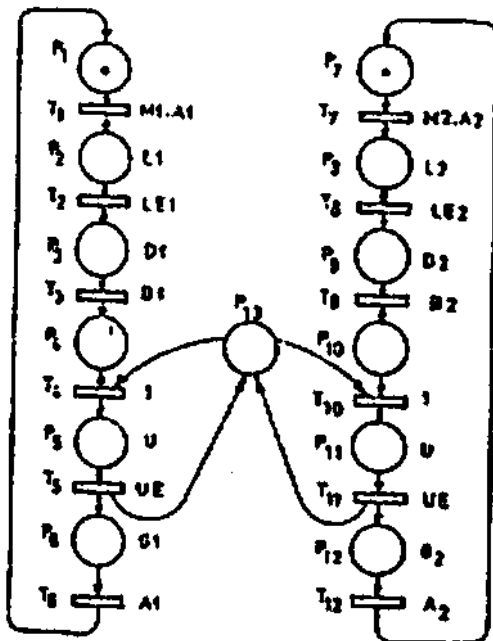
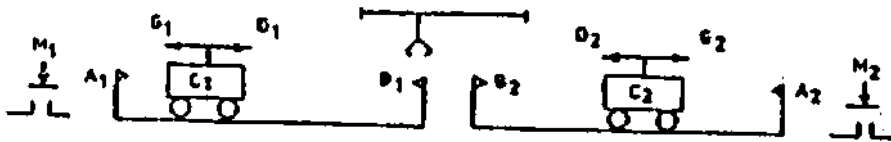
Místům nebo přechodům je přiřazen čas – přechod je uvolněn až ve chvíli, kdy tokeny v jeho vstupních místech setrvaly dostatečný čas

PS synchronizované

Uvolnění přechodu je podmíněno vnější událostí

Př.: Sdílení zdrojů, řešení pomocí PS

Sdílení zdrojů - pohyb dvou vozíků C1 a C2 po dvou drahách, pohyb řidičů G a D, start každého vozíku svým tlačítkem M. Vozík v A naložen a v B vyložen (sdílí vykládací mechanismus). V B čeká na obsazenou vykládku, automatický návrat do A po vyložení.



L_1 - naložení vozíku C_1

LE_1 - konec nakládání vozíku C_1

U - vykládání vozíku

UE - konec vykládání vozíku

P_{13} - vykládací mechan. (značený=volný)

Při obsazeném vykládacím mechanismu vozík čeká v B.

Literatura

- Přednášky DRS – Hanzálek, 1998
- Přednášky MKR – Bílek, 1997
- Kocourek P.: „Přenos informace“. Skripta. Vydavatelství ČVUT FEL, Praha 1996.
- Sylaby přednášek Bayer: „Logické řídicí systémy“. (Dobře vysvětleny Petriho sítě.) Soubory PCX na školní síti 1995.