TECHNOLOGIE PROJEKTOVÁNÍ INFORMAČNÍCH SYSTÉMŮ

Ing. Vladimír Pavlík

1. Úvod

V článku je vyložena původní koncepce technologie projektování aplikačních (informačních) systémů a vyvinuté programové prostředky pro realizaci aplikačních systémů na počítači.

Navržená technologie je založena na myšlence společné jednotné struktury všech deskriptivních jazyků pro popis aplikačních systémů. Deskriptivní (specifikační) jazyk je projektantský jazyk umožňující pohodlný, úplný a jednoznačný návrh uživatelských systémů dané třídy (spadajících do dané aplikační oblasti). Vytvořeno může být libovolné množství různých deskriptivních jazyků přesně přizpůsobených nejrůznějším potřebám rozmanitých aplikačních oblastí. Jednotná struktura deskriptivních jazyků je společná struktura všech deskriptivních jazyků nezávislá jak na specifice aplikační oblasti, tak i na druhu použité výpočetní techniky.

2. Jednotná struktura projektů aplikačních systémů

2.1 Celková charakteristika

Technologie projektování pracuje se dvěma základními pojmy: s pojmem projektového modulu a pojmem projektové sítě. Každá projektová jednotka - počínaje jednoduchým podprogramem a konče komplexem všech uživatelských algoritmů realizovaným jedním počítačem počítačové sítě - má jednotný tvar projektového modulu. Projektový modul představuje organickou jednotu algorit-

mických a paměťových funkcí, funkcí zpracování dat a funkcí řízení výpočtu. Projektová síť představuje jediný způsob, jak lze definovat nové, složené moduly na základě již definovaných modulů.

Projekt aplikačního systému je hierarchicky uspořádaná množina modulů. Hierarchická dekompozice algoritmů je ve vzájemně jednoznačném vztahu s dekompozicí údajové základny. To umožňuje přenositelnost libovolné části jednoho projektu (tj. modulu určitého hierarchického řádu) do libovolného jiného projektu a to bez nutnosti jakýchkoliv dodatečných úprav a to i tehdy, jedná-li se o přenos mezi různými typy výpočetní techniky.

Základní principy projektování jsou obecné a zcela nezávislé jak na typu použité výpočetní techniky, tak i na aplikační oblasti. Integrované programové prostředí pro vývoj projektů je vázáno jen na tyto principy, a je v tomto smyslu prázdné. Naplnění prázdného vývojového prostředí a tedy přizpůsobení specifickým vlastnostem konkrétní výpočetní techniky a specifickým potřebám konkrétní aplikační oblasti se děje pomocí dvou množin modulů: modulů programové báze a modulů projektové báze.

2.2 Projektové moduly

Pojem projektového modulu (dále jen modulu) je zobecněním všeobecně známého pojmu konečného automatu. Z abstraktně-teoretického hlediska je modul definován pěti množinami:

- množinou vstupních veličin
- množinou výstupních veličin
- množinou stavových veličin
- množinou vstupních bodů
- množinou výstupních bodů

a třemi funkcemi:

- výstupní funkcí
- stavovou funkcí
- funkcí přenosu řízení.

Vstupní, výstupní i stavové veličiny mohou být veličinami různých datových typů jednoduchých i složených (pole, záznamy). Vstupní (výstupní) body jsou body, kterými výpočetní proces do modulu vstupuje (z modulu vystupuje). Výstupní (stavová) funkce určuje nové hodnoty výstupních (stavových) veličin na základě hodnot vstupních veličin, předchozích hodnot stavových veličin a v závislosti na tom, v jakém vstupním bodě výpočetní proces do modulu vstoupil. Funkce přenosu řízení určuje výstupní bod na základě stejných argumentů. Poznamenejme, že libovolná z výše uvedených pěti množin může být i prázdná, a že množiny stavových a výstupních veličin se mohou prolínat (speciálně může být množina stavových veličin podmnožinou množiny vstupních veličin).

Graficky zobrazujeme moduly pomocí obdélníků; přitom: na levé (pravé) straně obdélníku znázorňujeme vstupní (výstupní) veličiny - datové vstupy (výstupy) a na horní (dolní) straně obdélníku znázorňujeme vstupní (výstupní) body - řídicí vstupy (výstupy). Stavové veličiny - nejsou-li současně výstupními veličinami - se nezobrazují. Ke všem vstupům a výstupům jsou uvnitř obdélníkového bloku uvedeny identifikátory. Uprostřed obdélníku je uveden identifikátor celého modulu.

Na obr. 1 a 2 jsou na ukázku zobrazeny moduly ADD a AROP pro přičítání reálných čísel a pro obecnou aritmetickou operaci s reálnými čísly. Funkce modulu ADD je následující: vstoupí-li výpočetní proces do modulu v bodě i (v bodě a), pak se provede přiřazení Y:=Y0 (Y:=Y+X) a výpočetní proces opustí modul ve stejnojmenném výstupním bodě. Modul AROP provádí přiřazení C=A∘B, kde význam operátorů ∘ (+,-, ,/,...) je dán hodnotou kódu na vstupu OP. Modul ADD má jedinou stavovou veličinu Y, která je současně veličinou výstupní; modul AROP nemá žádné stavové veličiny.

2.3 Projektové sítě

Pojem projektové sítě (dále jen sítě) je druhým základním pojmem popisované technologie projektování. Síť (nad danou množinou modulů) je definována třemi množinami:

- množinou příkazů
- množinou datových vazeb mezi příkazy
- množinou řídicích vazeb mezi příkazy

Každý příkaz sítě představuje volání některého modulu dané množiny. Datovou (řídicí) vazbou se rozumí spojení mezi datovým výstupem (výstupním bodem) jednoho příkazu a datovým vstupem (vstupním bodem) druhého příkazu. Zdůrazněme, že neexistují žádné jiné typy příkazů, než jsou volání modulů, a žádné jiné typy vazeb mezi příkazy, než jsou explicitně uvedena datová a řídicí spojení.

Na obr. 3 je na ukázku zobrazena síť pro výpočet kovariance

$$COV(X,Y) = N.\sum_{i=1}^{N} X_i.Y_i - \sum_{i=1}^{N} X_i . \sum_{i=1}^{N} Y_i$$

dvou veličin X,Y průběžně snímaných v reálném čase. Síť obsahuje devět příkazů, které volají moduly ADD a AROP popsané v kap. 2.2. Grafické zobrazení příkazů se od zobrazení volaných modulů liší jen tím, že v blocích jsou navíc uvedeny identifikátory příkazů (návěští), což jsou zpravidla přirozená čísla. Datové (řídicí) vazby mezi příkazy jsou vyznačeny plnými (přerušovanými) čarami.

S každým příkazem je pevně spojena jeho údajová základna tvořená hodnotami stavových a výstupních veličin. Každý příkaz může zapisovat pouze do své údajové základny, číst však může hodnoty výstupních veličin z údajových základen všech příkazů sítě..

Systematicky může být síť zadána tak, že postupně procházíme všechny příkazy sítě, a pro každý příkaz určujeme zdroje hodnot (cíle přenosu řízení) pro všechny datové vstupy (výstupní body). V případě datových vstupů mohou nastat tři případy:

- datovému vstupu je přiřazena konstanta (literál)
- zdrojem hodnoty pro datový vstup je datový výstup některého příkazu sítě (je-li datový výstup složeného datového typu, může být zdrojem hodnoty i jeho složka různé hierarchické hloubky)
- v rámci dané sítě není zdroj hodnoty pro datový vstup určen - takový datový vstup nazýváme volným v dané síti.

Pro výstupní body mohou nastat dva případy:

- cílem přenosu řízení je vstupní bod některého příkazu sítě
- v rámci dané sítě není cíl přenosu řízení pro výstupní bod určen - takový výstupní bod nazýváme volným v dané síti.

Pomocí sítí lze definovat nové složené moduly na základě již definovaných modulů takto:

- množina datových vstupů nového modulu je tvořena volnými datovými vstupy všech příkazů sítě; tato množina může být případně zmenšena ztotožněním některých datových vstupů téhož datového typu
- množina datových výstupů nového modulu je definována jako libovolně zvolená podmnožina množiny datových výstupů příkazů sítě
- množina stavových veličin nového modulu je sjednocení množin stavových veličin příkazů sítě
- množina vstupních bodů nového modulu je libovolně zvolená podmnožina množiny vstupních bodů příkazů sítě
- množina výstupních bodů nového modulu je tvořena volnými výstupními body příkazů sítě
- funkce nového modulu (výstupní, stavová, přenosu řízení) jsou jednoznačně určeny funkcemi modulů volanými v síti a datovými a řídicími vazbami mezi příkazy sítě.

Obdélník rámující síť na obr. 3 definuje modul COVAR pro průběžný výpočet kovariance dvou veličin. Tento modul může být použit v dalších sítích pro definici dalších složených modulů. Tak např. na obr. 4 je zobrazena síť a jí definovaný modul CORREL pro výpočet korelačního koeficientu:

$$COR(X,Y) = \frac{COV(X,Y)}{\sqrt{COV(X,X).COV(Y,Y)}}$$

2.4 Hierarchický systém modulů

Vztah mezi moduly spočívající v tom, že jeden modul slouží přímo nebo zprostředkovaně k definici jiného modulu, představuje částečné uspořádání na každé množině modulů. Vzhledem k tomuto uspořádání můžeme v každé množině modulů vydělit minimální moduly (nejsou definovány pomocí žádného jiného modulu dané množiny) a maximální moduly (neslouží k definici žádného jiného modulu dané množiny).

Absolutně minimální moduly (tj. moduly minimální v každé množině modulů) jsou tzv. elementární moduly, které nejsou definovány projektovými sítěmi, ale jsou naprogramovány v nějakém tradičním programovacím jazyce (C-jazyk, PASCAL). Absolutně maximální moduly (tj. moduly maximální v každé množině modulů) jsou tzv. uzavřené moduly, které mají prázdné množiny datových vstupů i výstupů, vstupních i výstupních bodů. Takové moduly nemohou být použity v žádné síti. Moduly, které nejsou elementární nazýváme složenými (strukturovanými), a moduly, které nejsou uzavřené nazýváme otevřenými.

Každému modulu můžeme přiřadit jeho hierarchický řád podle těchto pravidel:

- hierarchický řád elementárního modulu je roven nule
- hierarchický řád složeného modulu je o jedničku větší než je maximum z hierarchických řádů modulů volaných v jeho definiční síti.

Právě zavedené pojmy jsou ilustrovány na obr.5, na kterém je zobrazen diagram částečného uspořádání modulů ADD, AROP, COVAR a CORREL popsaných v předcházejícím textu.

Z pohledu technologie projektování aplikačních systémů definujeme 2 báze modulů:

- programová báze
- projektová báze

Programová báze je tvořena výhradně elementárními moduly a slouží k zakrytí specifik technických prostředků a základního programového vybavení rozmanitých konkrétních počítačů. Vytváří tak jakousi počítačově nezávislou bázi pro budování rozličných projektových jazyků. Programová báze je implementována profesionálními programátory.

Projektová báze je tvořena jak elementárními, tak i složenými moduly, které definují projektový (deskriptivní) jazyk pro určitou konkrétní aplikační oblast. Projektová báze je vytvářena pomocí prostředků jednotné struktury deskriptivních jazyků, a proto její implementace, kromě programování speciálních elementárních modulů, nevyžaduje profesionálního programátora, ale zkušeného projektanta v dané aplikační oblasti.

3. Programové prostředí pro vývoj a provoz aplikačních systémů

Pro technologii projektování aplikačních systémů, která byla stručně charakterizována ve 2. kapitole, je vyvíjen programový systém na počítači typu IBM PC.

Programové prostředí pro vývoj a provoz aplikačních systémů je vytvořeno jako integrované vývojové prostředí s následujícími systémovými programy:

- řídicí program (supervizor) programového prostředí
- interaktivní editor pro vkládání projektu do počítače
- programy pro analýzu, zobrazení a transformaci projektu
- překladač projektu do interpretovatelné formy
- ladicí program modulů
- operačně-interpretační systém

Řídicí program zabezpečuje spouštění a ukončování ostatních systémových programů (výběr z nabídky základních funkcí integrovaného vývojového prostředí).

Interaktivní editor slouží ke vkládání projektu do počítače. Toto vkládání je prováděno po elementárních projektových údajích tak, že každý elementární údaj je vložen pomocí editoru do vstupního pole, a pak je testován na syntaktickou správnost a konzistenci s dříve zadanými údaji. Jedině správný údaj je uložen do datových struktur, které tvoří zápis projektu na počítači. Interaktivní editor má 4 režimy:

- definování datových typů (pole, záznamy)
- definování vnějších popisů modulů
- definování vnitřních popisů modulů (sítí složených modulů)
- definování rozložení projektu na jednotlivé počítače.

Vnitřní popisy modulů (sítě) jsou zadávány v grafickém jazyce (viz obr. 1 až 4). Ostatní údaje jsou definovány v textovém jazyce a mohou být doplněny komentáři.

Programy pro analýzu, zobrazení a transformaci projektu slouží projektantovi k provádění různých pomocných funkcí, jako např.:

- nalezení všech minimálních (maximálních) modulů
- očistění projektu od nepotřebných modulů a datových typů
- zobrazení hierarchie volání modulů v projektu
- zjištění volných datových vstupů (výstupních bodů)

Překladač provádí transformaci projektu, který byl vložen do počítače pomocí interaktivního editoru, do interpretovatelné formy. Jedná se buď o zápis projektu v C-jazyce, který může být spojen s interpretačním programem a knihovnou elementárních modulů ručně (a to i na jiném počítači), nebo se jedná o spustitelný program na počítači IBM PC.

Ladicí program ověřuje funkci části projektu - projektového modulu (elementárního nebo složeného), který byl přeložen překladačem. Ladicí program zastavuje interpretaci modulu po každém průchodu, zobrazuje hodnoty vstupních a výstupních veličin a vstupní a výstupní body, nastavuje hodnoty vstupních veličin a vybírá vstupní bod.

Operačně-interpretační systém je program, který provádí interpretaci projektu. Projekt je v tomto případě tvořen datovými strukturami, které obsahují datové a řídicí tabulky a odkazy na kódy elementárních modulů.

4. Závěr

Pro nedostatek místa článek podává neúplný a místy i zjednodušený výklad jinak široké problematiky projektování in-

formačních systémů na uvedených principech. Uvedený triviální příklad pouze ilustruje základní principy.

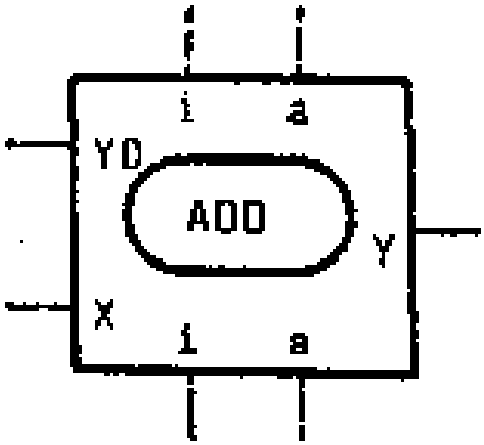
Nová technologie projektování bude dostupná na konci roku 1989, kdy bude ukončeno řešení v rámci státního úkolu a k dispozici bude první verze experimentálního programového vybavení pro vývoj a provozování řídicích systémů bez vytvoření projektových bází pro určité aplikační oblasti.

Literatura

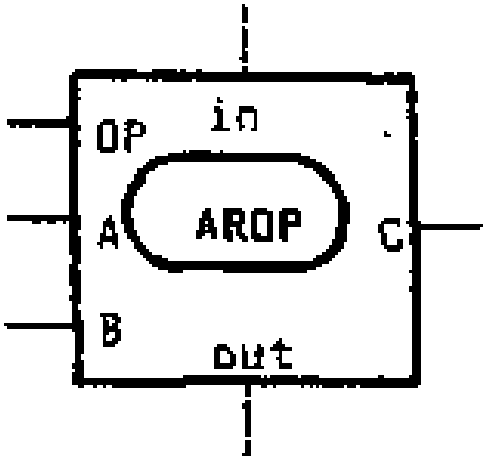
/1/ Markl,J.: Technologija projektirovanija informacionnych sistem i ich programnoj realizacii. INTERFACE RVHP - konkursní práce, 1986

/2/ Markl,J.,Batková,I.,Růžičková,P.: Jednotná struktura deskriptivních jazyků (Úvodní studie). Hanoclášské stroje Ostrava, 1987

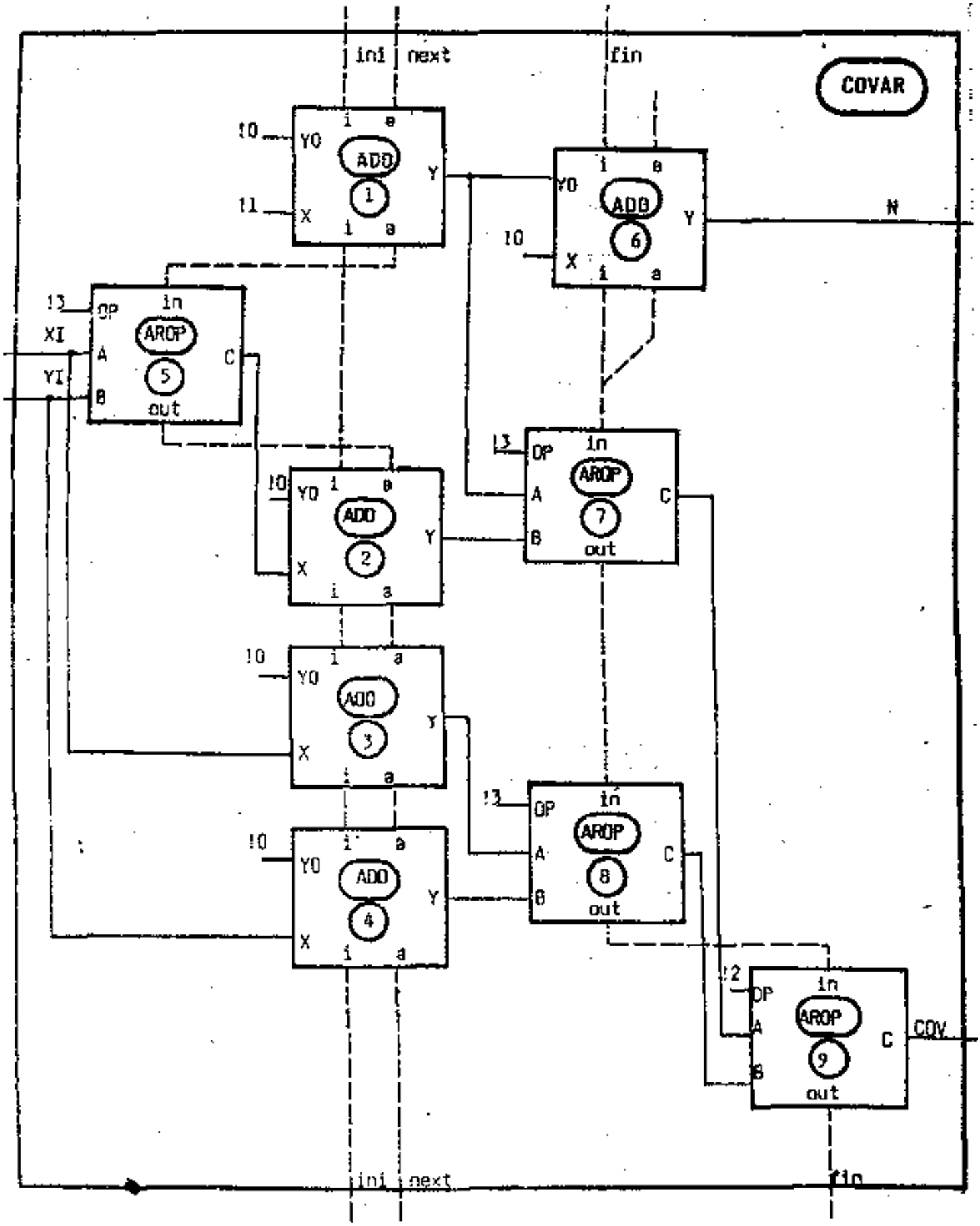
/3/ Markl,J.,Batková,I.,Beyer,A.: Jednotná struktura deskriptivních jazyků (Projekt systémových prostředků). Energa Ostrava, 1988



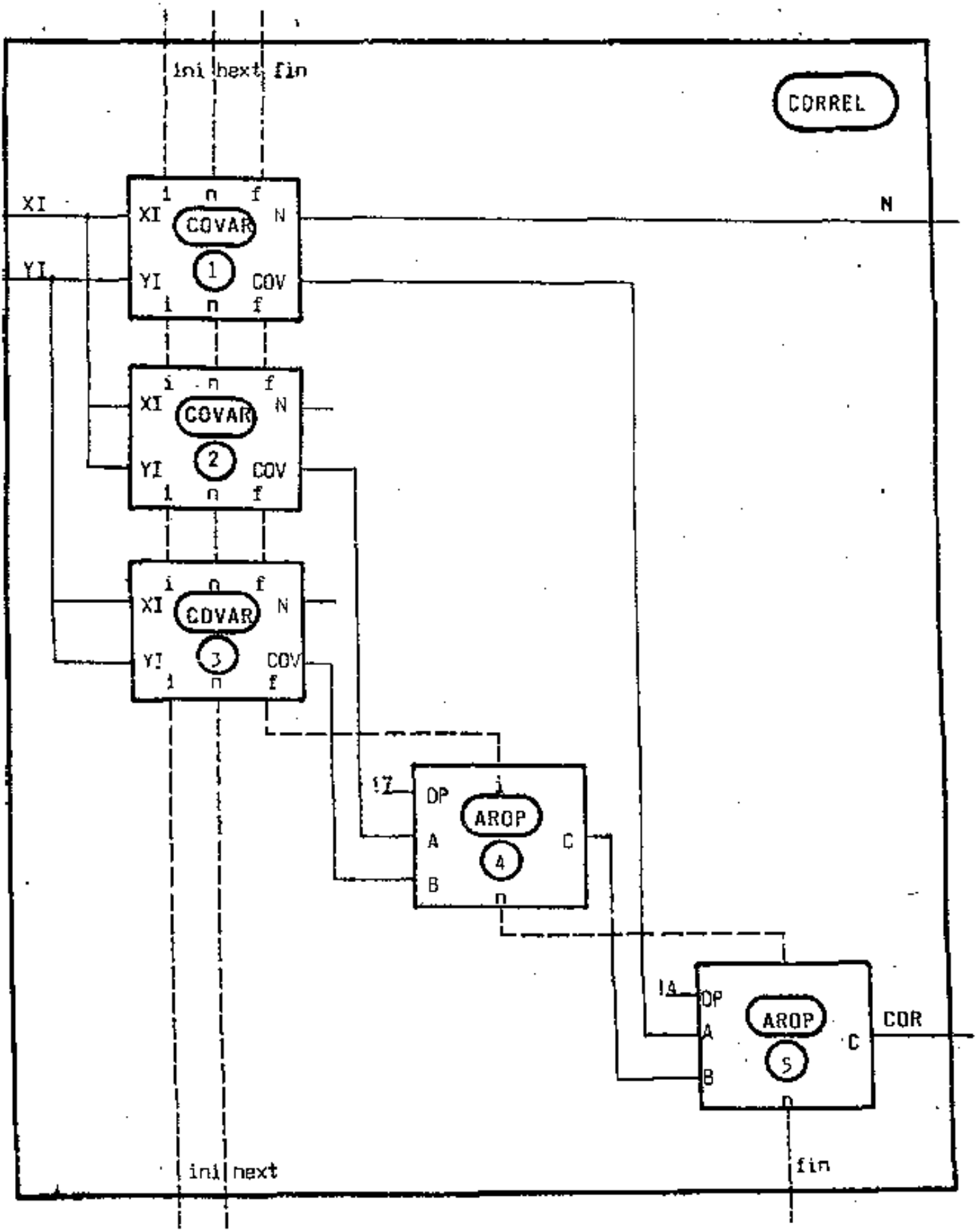
Obr. 1. Modul ADD



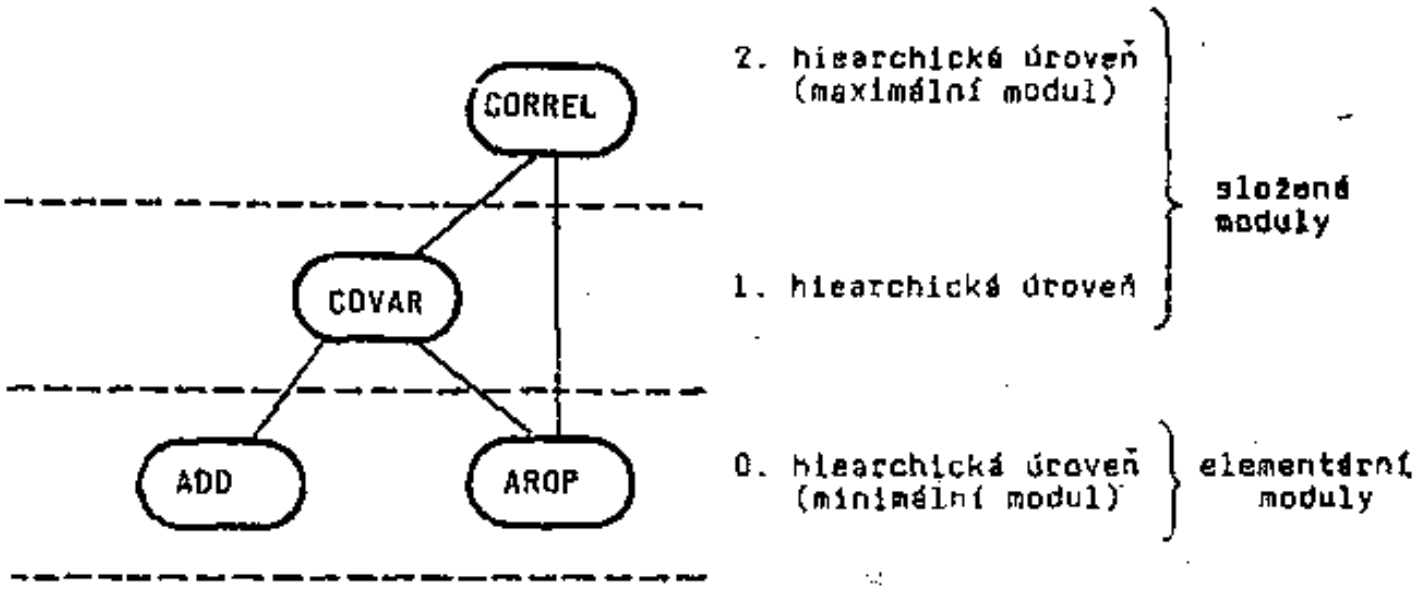
Obr. 2. Modul AROP



Obr. 3. Definiční síť modulu COVAR



Obr. 4. Definiční síť modulu CORREL



Obr. 5. Částečné uspořádání modulů