

DIPLOMOVÝ ÚKOL

s. Stanislava B e r á n k a

pro

obor **automatizované systémy řízení výrobních procesů ve strojírenství**

Protože jste splnil..... požadavky učebního plánu, zadává Vám vedoucí katedry ve smyslu směrnice ministerstva školství ~~xxxxxx~~ o státních závěrečných zkouškách tento diplomový úkol:

Název tématu: **Algoritmy systému logického řízení mlecích okruhů bloku 200 MW**

Pokyny pro vypracování:

1. Prostudujte z literárních pramenů a podkladů EGÚ technické podmínky provozu mlecích okruhů elektrárenských bloků a způsoby jejich řízení.
2. Proveďte návrh algoritmů logického řízení mlecích okruhů. Volte vhodný zápis algoritmů (vývojový diagram, logické funkce, rozhodovací tabulky).
3. Realizujte návrh logického řízení na počítači RPP 16 a ověřte jeho funkci simulací.
4. Seznamte se s programovacím systémem KORAL a přepište vybraný algoritmus pro konkrétní elektrárnu do tohoto systému.
5. Proveďte zhodnocení návrhu řízení v porovnání s obvody pevné logiky.

Autorské právo se řídí směrnicemi
MŠK pro státní záv. zkoušky č.j. 31
727/62-III/2 ze dne 13. července
1962-Věstník MŠK XVIII, sešit 24 ze
dne 31.3.1962 a č. 115/53 Sb.

V 432/1/81 S
VYSOKÁ ŠKOLA STROJNÍ A TEXTILNÍ
Univerzitní knihovna
LIBEREC, PŘÍRODOVĚDECKÁ 5
742 01

Rozsah grafických laboratorních prací:

Rozsah průvodní zprávy:

50 - 60 stran

Seznam odborné literatury:

Hoernes, G.E. - Heilweil, M.F.: Úvod do Booleovy algebry a navrhování logických obvodů. Praha, SNTL 1961

Chvalkovský, V.: Rozhodovací tabulky. Praha, SNTL 1976

Pedklady EGÚ (výzkumné zprávy, programy, projekty)

Vedoucí diplomové práce:

Prof. Ing. Bořivoj H a n u š, CSc.

Konsultanti:

Ing. Bohumil P i d e r m a n n, CSc. - EGÚ Praha

Ing. Bedřich J a n e č e k

Datum zahájení diplomové práce:

15.9.1980

Datum odevzdání diplomové práce:

12.6.1981



Alaxin
Doc. Ing. J. Alaxin, CSc.

Vedoucí katedry

Stříž
Doc. RNDr. B. Stříž, CSc.

Děkan

Vysoká škola strojní a textilní v Liberci
nositelka Řádu práce

Fakulta strojní

Obr 23 - 40 - 8
automatizované systémy řízení výrobních procesů
ve strojírenství

Matouš technické kybernetiky

Algoritmy systému logického řízení elektrických obvodů
elektr. 220 MW

Stanislav BURÁNEK

Vedoucí práce: Prof. Ing. Bořivoj HANUŠ, CSc. - VŠST Liberec

Konzultant: Ing. Bohumil FIDERMANN, CSc. - EGÚ Praha

HKH ASČOP - 001

Režisér práce a příloh:

Počet stran : 54
Počet příloh: 3
Počet tabulek: 27

V Liberci 12. června 1981

Místopřisežně prohlašuji, že jsem diplomovou práci
vypracoval samostatně s použitím uvedené literatury.

V Liberci 12. června 1981

Stanislav BERÁNEK
Beránek

Vysoká škola strojní a textilní v Liberci
nositelka Řádu práce

Fakulta strojní

Obor 23 - 40 - 8
automatizované systémy řízení výrobních procesů
ve strojírenství

Katedra technické kybernetiky

Algoritmy systému logického řízení mlecích okruhů
bloku 200 MW

Stanislav BERÁNEK

Vedoucí práce: Prof. Ing. Bořivoj HANUŠ, CSc. - VŠST Liberec

Konzultant: Ing. Bohumil PIDERMANN, CSc. - EGÚ Praha

KTK ASŘSF - 001

Rozsah práce a příloh:

Počet stran : 54
Počet příloh: 6
Počet tabulek: 27

276/4

V Liberci 12. června 1981

OBSAH

SEZNAM POUŽITÝCH ZKRATEK

1. ÚVOD

1.1 Obecně o ASŘ

1.2 ASŘ TP v elektrárnách

2. ŘÍZENÍ MLECÍCH LINEK

2.1 Současný stav v řízení mlecích linek

2.2 Technické podmínky provozu algoritmu skupiny spalování

2.2. 1 Konkrétní technické podmínky

3. NÁVRH LOGICKÉHO ŘÍZENÍ

3.1 Jednotlivé subalgoritmy

3.3 Rozhodovací tabulky jako vhodný nástroj zápisu logického řízení

3. 3.1 Vývojové diagramy a rozhodovací tabulky

3.3.2 Strukturní prvky a druhy rozhodovacích tabulek

3.3.3 Zmenšování rozsahu rozhodovacích tabulek

3.3.4 Dělení a spojování rozhodovacích tabulek

3.3.5 Převod rozhodovacích tabulek do programu

3.3.6 Automatizovaný převod rozhodovacích tabulek do programu

4. POPIS PROGRAMU

4.1 Převod rozhodovacích tabulek K3-4 do programu

4.2 Popis uživatelského programu

4.3 Popis simulačního programu

4.4 Popis simulačních zkoušek

5. ALGORITMUS K3-4 V JAZYKU KORAL

5.1 Seznam použitých příkazů

5.2 Program v jazyku KORAL

6. ZHODNOCENÍ NÁVRHU ŘÍZENÍ A POROVNÁNÍ S OBVODY PEVNÉ LOGIKY

7. ZÁVĚR

LITERATURA

PŘÍLOHY: 1 - SCHEMA SOUČINNOSTI VÝVOJOVÝCH DIAGRAMU K3

2 - VÝVOJOVÝ DIAGRAM K3-4

3 - ROZHODOVACÍ TABULKY K3-4

4 - UŽIVATELSKÝ PROGRAM

5 - SIMULAČNÍ PROGRAM

6 - SIMULAČNÍ ZKOUŠKY

SEZNAM POUŽITÝCH ZKRATEK

ASŘ TP - automatizovaný systém řízení technologického procesu

VHJ - výrobně hospodářská jednotka

JSEP - jednotný systém elektronických počítačů

SMEP - systém malých elektronických počítačů

RT - rozhodovací tabulky

ML - mlecí linka

skupinový automat skupiny spalování

K

MH - mazutový hořák

K 31 - motor mazutových hořáků

K 32 - motor ML 1

K 33 - motor ML 2

K 34 - motor ML 3

K 35 - motor ML 4

K 36 - motor ML 5

K 37 - motor ML 6

Dvouhodnotové vstupy do algoritmu skupiny spalování z procesu

OV 1 - omezit výkon

QZH 3 - možno najet MH

QZL 4 - požadavek přiřazení ML

SKL 1 - podmínka počátku protáčení turbíny splněna

MHZ 0 - K31 najeto

SVP 1 - vstupní podmínka K 32 ML 1

MLZ 1 - K 32 najeto

AOP 1 - autonomní ochrana K 32 působí

MZK 1 - motor K 32 zapnut

SVP 2 - vstupní podmínka K 33 ML 2

NLZ 2 - K 33 najeto

MZK 2 - motor K 33 zapnut
 SVP 3 - vstupní podmínky K 34 ML 3
 MLZ 3 - K 34 najeto
 AOP 3 - autonomní ochrana K 34 působí
 MZK 3 - motor K 34 zapnut
 SVP 4 - vstupní podmínky K35 ML 4
 MLZ 4 - K 35 najeto
 AOP 4 - autonomní ochrana K 35 působí
 MZK 4 - motor K 35 zapnut
 SVP 5 - vstupní podmínky K 36 ML 5
 MLZ 5 - K 36 najeto
 AOP 5 - autonomní ochrana K 36 působí
 MZK 5 - motor K 36 zapnut
 SVP 6 - vstupní podmínky K 37 ML 6
 MLZ 6 - K 37 najeto
 AOP 6 - autonomní ochrana K 37 působí
 MZK 6 - motor K 37 zapnut
 MHZ 1 ÷ MHZ 6 - MH 1 + MH 6 najet
 AML 1 + AML 6 - ML 1 + ML 6 řízeno z K 3

Dvouhodnotové výstupy z algoritmu skupiny spalování do procesu

HL 17 - nepřípustná konfigurace
 ZNK 1 - není uvolnění z automatu počátku protáčení turbíny
 ZNK 2 - není žádán výkon 12 %
 NAL 1 + NAL 6 - K 32 + K 37 najet

Logické rovnice K 3

L 10 = MZK 1 x MZK 2 x MZK 3 x MZK 4 x MZK 5 x MZK 6
 L 21 = AML 1 x QZL 4 x SVP 1 x - / MLZ 1 + AOP 1 /
 L 22 = AML 2 x QZL 4 x SVP 2 x - / MLZ 2 + AOP 2 /
 L 23 = AML 3 x QZL 4 x SVP 3 x - / MLZ 3 + AOP 3 /

L 24 = AML 4 x QZL 4 x SVP 4 x - / MLZ 4 + AOP 4/
 L 25 = AML 5 x QZL 4 x SVP 5 x - / MLZ 5 + AOP 5/
 L 26 = AML 6 x QZL 4 x SVP 6 x - / MLZ 6 + AOP 6/
 L 42 = SKL 1 x QZH 3 x NHK 3 x - / AOP 1 + AOP 2 + AOP 3 + AOP 4 +
 + AOP 5 + AOP 6 / + QZL 4 x PA 1 x - / AOP 1 + AOP 2 +
 AOP 3 + AOP 4 + AOP 5 + AOP 6 / + QZL 4 x PA 2 x - / AOP 1
 + AOP 2 + AOP 3 + AOP 4 + AOP 5 + AOP 6 /

Časové členy

- A 1 - limit najetí mazutových hořáků
- A 2 - Trvá-li po najetí mlecího okruhu požadavek na najetí dalšího, pak se čeká
- A 18 - limit pro rozhodnutí obsluhy zda ovládání K3 aut. nebo ručně
- A - limit najetí mlecích linek

Vnitřní proměnné

- PA 1 - nabývá hodnotu 1, když při splnění vstupní podmínce L 42 je počet najetých mlecích okruhů 3. Zabráňuje, aby při startu ze studeného stavu nebyl program K3-4 spuštěn pouze požadavkem na zvýšení příkonu tepla. Maže se při odstavení tlačítkem STOP.
- PA 2 - Nabývá hodnotu 1, když při splněné vstupní podmínce L 42 jsou najety mazutové hořáky. Zabráňuje, aby při startu ze studeného stavu nebyl program K3-4 spuštěn pouze požadavkem na zvýšení příkonu tepla. Maže se při odstavení tlačítkem "STOP".
- PA 3 - Nabývá hodnotu 1, když nedojde k najetí mazutových hořáků a tím se blokuje pokus v dalším cyklu o jejich znovunajetí. Maže se stisknutím tlačítka "START".

- C 9/ L/ - Výstupní proměnná C9 umožňuje ve spolupráci se skokem řízeným měnit místo vstupu do programu K3-4
- C 11 - reprezentuje vypnutý mlecí okruh č. 1
- C 12 - reprezentuje vypnutý mlecí okruh č. 6
- C 13 - reprezentuje vypnutý mlecí okruh č. 3
- C 14 - reprezentuje vypnutý mlecí okruh č. 4
- C 15 - reprezentuje vypnutý mlecí okruh č. 2
- C 16 - reprezentuje vypnutý mlecí okruh č. 5
- P 7 - součet zapnutých mlecích okruhů levé strany spalov.komory
- P 8 - součet zapnutých mlecích okruhů pravé strany spalovací komory
- C 32 - součet zapnutých mlecích okruhů pravé a levé strany spalovací komory
- C 30 - rozdíl zapnutých mlecích okruhů levo a pravé strany spalovací komory
- C 6 - udává postupně číslo místa vypnutého mlecího okruhu

1 Úvod

Usnesení XVI sjezdu zdůraznilo význam racionalizace řízení a technického rozvoje, účinné zavádění výsledků vědecko-technické revoluce do praxe a to zvláště ve strojírenství, které je nosným programem v československém národním hospodářství. Jednou z možností realizace těchto úkolů je důsledné zavádění automatizace formou automatizovaných systémů řízení na všech úrovních.

1. 1 Obecně o ASŘ

Systém řízení v podmínkách československého národního hospodářství je založen na odvětvovém principu. Hierarchickou strukturu lze pak rozdělit na čtyři úrovně. Nejvyšší úroveň je odvětvový automatizovaný systém řízení, jež působí v rámci celého resortu. Nižší stupeň je na úrovni VHJ. Jde o ASŘ středního článku. Další nižší úrovní je ASŘ v podniku, ve kterém jsou soustředěny automatizované funkce, vykonávané na úrovni podniku. Nejnižší úrovní je ASŘ výrobního procesu a technologického procesu, které se týkají přímo procesu výroby. A právě na tuto nejnižší úroveň je třeba se nejvíce zaměřit a pomocí automatizační techniky snižovat pracovní zátěž a zvyšovat produktivitu práce. Ovšem pro splnění těchto úkolů je třeba mít potřebné technické vybavení. Cenová nedostupnost a rozličné množství typů techniky zatím bránilo ve větším rozšíření ASŘ VP. Avšak rozvojem výpočetní techniky řady JSEP a SMEP v rámci RVHP dojde k typovému sjednocení, dostupnějším cenám a díky tomu také k uplatnění spolupráce mezi jednotlivými podniky, zvláště v softwarové dokumentaci.

1. 2 ASŘ TP v elektrárnách

Jedním ze základních odvětví národního hospodářství ČSSR je energetika. Proto je velmi potřebné, aby výroba energie byla

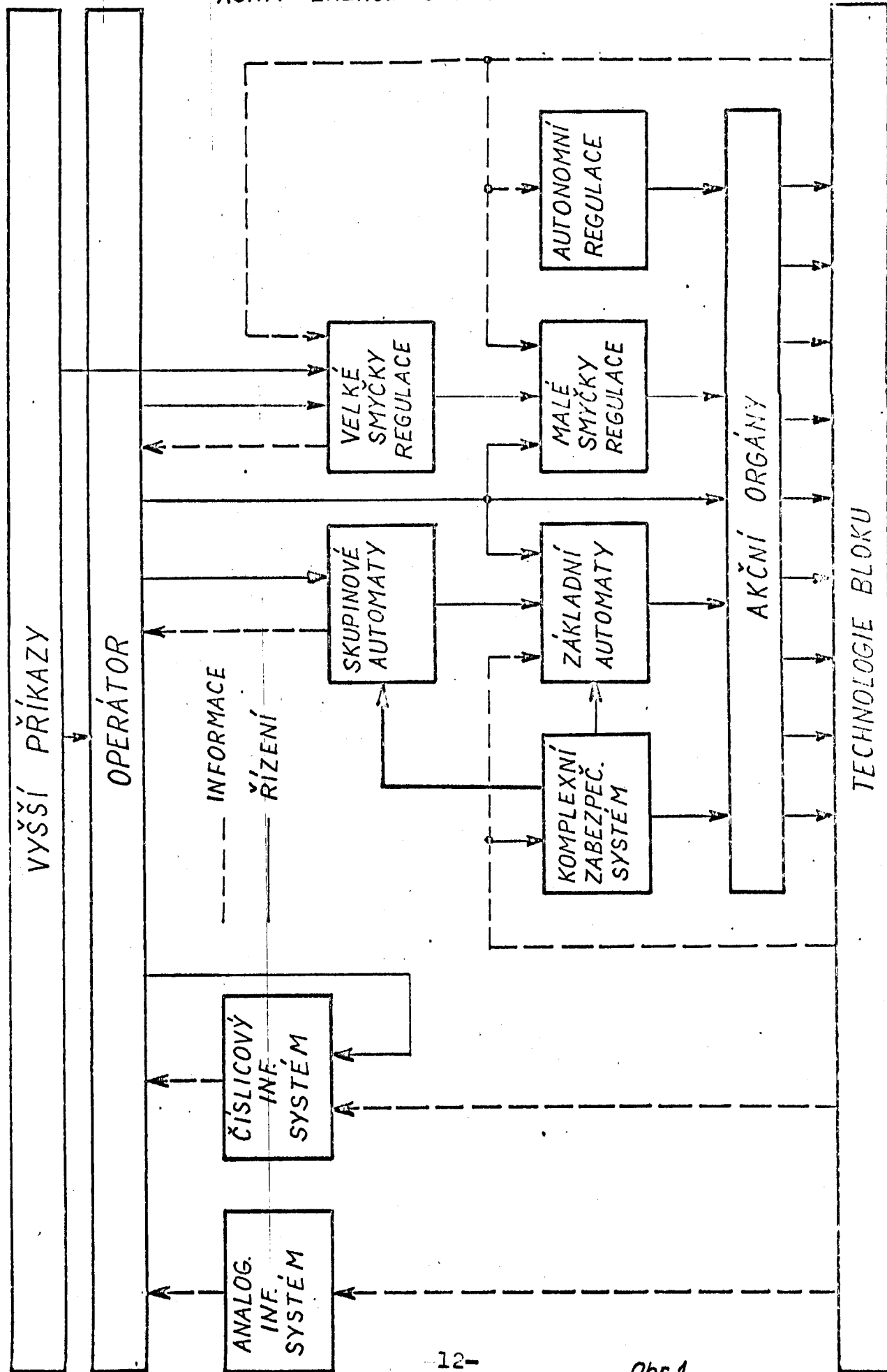
co nejefektivnější. Toho v dnešní době nelze dosáhnout jinak, než zavedením ASŘ a to ve všech úrovních. Výroba elektrické energie je v elektrifikační soustavě ČSSR zajišťována převážně tepelnými elektrárnami, které můžeme dělit z hlediska stupně automatizace na dvě kategorie. Charakteristickým rysem kategorie A patří omezený rozsah funkcí informačního a řídicího systému z převládajícím ručním řízením, uplatnění regulačních okruhů především v regulovaném rozsahu, ojedinělá automatizace ovládnutí některých funkčních celků a omezený rozsah funkcí zabezpečovacího systému. Kategorie B se vyznačují hierarchickým uspořádáním umožňujícím řízení bloku v několika úrovních, informačním systémem zajišťovaným počítačem s koncentrací informace obsluhy do několika málo sdělovačů, automatickým najížděním a odstavováním bloku, působením regulačních okruhů za všech provozních stavů a komplexním zabezpečovacím systémem.

Automatizaci ASŘ TP lze rozdělit do tří skupin.

- S 5 - Akční orgány jednotlivých částí funkčních celků, jako např. MLL spínače ..
- S 4 - Základní automaty - zabezpečují automatický provoz celých funkčních celků / napáječka, ML /.
- S 5 - Skupinové automaty - zabezpečují provoz celých funkčních skupin / spalování / schema hierarchického uspořádání je na obr. 1.

Pro projektové využití je potřebný vhodný zápis daného logického řízení. Jde tedy o to dělat co nejjednodušší vývojové diagramy. Ovšem nejpřehledněji plní tuto funkci rozhodovací tabulky. Proto jsem se při zpracování logického řízení skupiny spalování zaměřil zvláště na jejich využití a pomocí nich také k zjednodušení programu. Tento přístup by měl tedy zpřístupnit řízení elektrárenských bloků i pro pracovníky, kteří nejsou

ASŘTP ENERGETICKÉHO BLOKU 200 MW



plnými odborníky, přesto však pracují s jejími výsledky a zkrá-
tit strojní čas potřebný k výpočtu programu. Podle diplomového
zadání jsem se zaměřil na funkční skupinu spalování, zvláště
na vlastní spouštění a odstavování mlecích linek v provozu i
po havárii.

2 Řízení mlecích linek

2. 1 Současný stav v řízení mlecích linek

V současné době se jednotlivé tepelné elektrárny od sebe
odlišují jednak druhem mlecích linek, jednak počtem. V katego-
rii A se objevují tlukadlové mlýny, které nejsou plně automaticky
řízeny. Avšak trend v ČSSR spěje ke kategorii B. Zde se uplat-
ňují výhodnější a modernější ventilátorové mlýny. U největšího
elektrárenského bloku v ČSSR EMĚ III , je těchto mlýnů deset.

Logické řízení okruhů v elektrárnách je u naprosté většiny
realizováno pomocí automatů pevné logiky. Tyto jsou však velmi
rozměrné a nedynamické. Naproti tomu v moderní elektrárně EMĚ III
je již řízení mlecích linek zajištěno analogo-číslicovými prog-
ramovatelnými automaty ZEPALOG / výrobek národního podniku ZPA /.
Ale ve všech elektrárnách v ČSSR součinnost jednotlivých celků
zajišťuje operátor z dozorny.

2. 2 Technické podmínky provozu algoritmu skupiny spalování.

K nejobtížnějším úlohám při návrhu projektových podkladů
náleží získání informací a podmínkách provozu řízeného objektu.
Zde je třeba brát v úvahu názory a návrhy projektanta, zvláště
v případě samostatné projektové organizace, výrobce technolo-
gického zařízení a v neposlední řadě provozovatele, který dří-
vější teoretické názory zásadně koriguje reálnými možnostmi
a potřebami řízení technologického zařízení.

Ani zde nelze hovořit o konečném tvaru řídicího algorit-
mu, protože teprve dlouhodobý provoz a případné změny v důsledku

průběžných a generálních oprav ověří správnost návrhu.

2. 2. 1 Konkrétní technické podmínky

Pro spuštění algoritmu skupiny spalování je nezbytné, aby platily vstupní podmínky start, t.j.

- na obou stranách spalovací komory je spuštěno po dvou mlecích linkách nebo / při startu ze studeného stavu / jsou splněny vstupní podmínky startu těchto linek,
- jsou splněny podmínky najetí mazutových hořáků,
- je najeta skupina napájení,
- z hlavy algoritmu tlačítkem na hlavním pultu operátora je dán povel ke startu,
- pro hlášení, že skupina spalování je najeta postačí, když je najeta některá z mlecích linek,
- pro hlášení, že skupina spalování je odstavena je zapotřebí, aby byly vypnuty zdroje tepla.

Při startu ze studeného stavu je zapotřebí nejprve zapnout skupinu mazutových hořáků. Podmínky startu mazutových hořáků jsou:

- jsou splněny podmínky startu algoritmu skupiny spalování,
- požadovaný příkon tepla je větší jak 12%,
- jsou splněny podmínky pro najetí zdrojů tepla na funkčním celku turbosoustrojí.

Při startu funkční skupiny spalování za provozu, t.j., že motor některé mlecí linky je již najet, nedojde ke startu mazutových hořáků a za předpokladu, že existuje požadavek na zvýšení příkonu tepla, je možné odstartovat mlecí linku.

Najetím mazutových hořáků, nebo splněním podmínek ke startu mlecí linky vzniká trvalý signál, umožňující zapnutí další mlecí linky, a to v případě, že vznikne požadavek na zvýšení příkonu tepla. Tento signál se zruší až po vydání povelu k odstavení.

Dříve než dojde ke startu mlecí linky, je třeba její zapnutí prověřit z hlediska vhodnosti konfigurace. Za vhodnou konfiguraci se pokládá stav, kdy rozdíl mezi počtem zapnutých mlecích linek pravé a levé strany spalovací komory není větší než jedna. Za nevhodnou konfiguraci je považován stav, kdy tento rozdíl je roven dvěma. Pokud je rozdíl roven třem, je konfigurace nepřipustná. Nevhodnou konfiguraci lze zapnout pouze tehdy, když není k dispozici žádná vhodná.

Mlecí linky je třeba zapínat v pořadí 6-1-4-2-5-3.

Ke startu mlecí linky je zapotřebí, aby byly splněny tyto vstupní podmínky:

- trvá požadavek na zvýšení příkonu tepla,
- je zapnut mazutový hořák zapínané linky nebo jsou najety aspoň další dvě mlecí linky,
- zapnutí způsobí v krajním případě nevhodnou konfiguraci,
- jsou splněny podmínky startu mlecí linky a její automat na IV. úrovni je přepnut do polohy " automaticky,"
- start se neuskuteční, pokud je již mlecí linka najeta, nebo působí její autonomní ochrana, anebo neuplynula nejméně jedna minuta od najetí předchozí mlecí linky.

Pro odstavení mlecí linky je zapotřebí, aby

- existoval požadavek na snížení příkonu tepla,
- konfigurace při odstavení mlecí linky byla v krajním případě nevhodná,
- automat mlecí linky na IV. úrovni byl přepnut do polohy " automaticky,"
- uplynula alespoň jedna minuta od okamžiku, kdy zanikl signál " ML najeta " u té mlecí linky, která byla odstavována předtím.

Další mlecí linka je startována tehdy, když trvá požadavek na zvýšení tepelného příkonu a přitom nejsou splněny požadavky

pro start předcházející mlecí linky.

Naopak další mlecí linka je odstavována, pokud trvá požadavek na snížení tepelného příkonu a přitom předchozí linka nebyla zapnuta do automatického provozu, nebo jejím odstavením by nebyly splněny požadavky vhodnosti konfigurace, nebo byla odstavena.

Obsluhu je třeba informovat o stavu automatu světelnými návěstími ve hlavě na panelu operátora. Jsou to stavy " najíždí-najeto", "odstavuje - odstaveno" , "splněné podmínky ", "nepřípustná konfigurace", "není žádán výkon 12%" a " není uvolnění startu z S 11 ".

- Algoritmus K 3 umožňuje, že jeho start lze provést tlačítkem " START " rovněž za provozu bloku, tedy ne pouze z počátečního stavu při vypnutých zdrojích tepla. To bylo umožněno vložením podmínky, která zjišťuje kolik mlecích linek je v provozu. Pokud jejich počet je roven nebo menší než tři, jsou zapnuty mazutové hořáky; v opačném případě, za předpokladu existence požadavku na zvýšení příkonu tepla, je zapnuta mlecí linka.
- Je umožněn zásah obsluhy operátora. Pokud je kontakt tlačítka zapnut, pak žádnou z mlecích linek zapojenou do nadřazené logiky K 3 nelze automaticky ovládat počítačem, obsluha tak má prioritu nad činností počítače. Přitom počítač sleduje činnost obsluhy, což umožňuje přepnutí prakticky v libovolném okamžiku na automatický provoz.
- Je-li některý z mlecích okruhů vyřazen pro poruchu svoji autonomní ochranou, pak je zapotřebí pro dodržení žádané hodnoty výkonu ihned zapnout další mlecí okruh. Přitom v přechodném období, než mlýn najede na výkon, jsou zapnuty mazutové hořáky.
- Dojde-li v důsledku havárie / např. poruchy kouř. ventilátoru /

- k výpadku všech zdrojů tepla, pak je výhodné zapnout ty mlýny, které byly v okamžiku havarie v provozu, tedy mlýny teplé.
- Pro vstupní podmínky namísto logického součtu zapnutých elektronapáječek a turbonapáječky postačí, aby byl sledován průtok napájecí vody.
 - Pro Podmínku, že algoritmus K 3 je najet, nestačí, že je najet alespoň některý mlecí kruh. Je třeba, aby současně neexistoval požadavek na přidání nebo ubrání mlecího okruhu / L 60 /.
 - Podmínka, že mlecí okruh je zapnut, je dán logickým součinem motor zapnut X vrstva uhlí na podavači.
 - K prověřování vstupních podmínek ke startu zvoleného mlecího okruhu dojde pouze tehdy, když je v provozu jeho mazutový hořák, anebo jsou v provozu nejméně tři další mlecí okruhy.
 - Havarijní odstavení všech zdrojů tepla tlačítkem "STOP" je možné tehdy, až výkon bloku je snížen na minimum, t.j. 100MW.
 - Při havarijním odstavení všech zdrojů tepla tlačítkem "STOP" jsou mazutové hořáky odstaveny až 1 minutu po odstavení mlecích okruhů.
 - Stav " nepřípustná konfigurace " způsobí zablokování činnosti algoritmu pro přidání nebo ubrání mlecího okruhu. Odblokování provede obsluha po odstranění tlačítkem "START".
 - Pokud mazutové hořáky nenajedou při startu v daném limitu, je algoritmus zablokován a jeho odblokování provede obsluha opět tlačítkem "START".

3 Návrh logického řízení.

Logické řízení skupiny spalování bylo předmětem práce Ing. B. Pidrmanna. Obsahem mé práce bylo prostudovat technické podmínky skupiny spalování a algoritmi, které tuto skupinu řídí. Hlavním přínosem této práce je nové řešení algoritmu K 3. 4 metodou vývojového diagramu, zápis algoritmu pomocí rozhodovacích

tabulek a programové řešení vycházející ze zásad přepisu RT do programovacího jazyka.

Návrh logického řízení byl dělán z posledních technických podmínek. Po dohodě s Ing. B. Pidermannem jsem zachoval původní rozdělení a součinnost algoritmů.

Vlastní algoritmus číslicové části skupiny spalování byl rozdělen do osmi subalgoritmů, a to hlava K3-1, zrušení povelů start-stop K3-2, odstavení všech zdrojů tepla do studeného stavu K3-3, start mazutových hořáků a mlecích linek, jakož i zapínání dalších mlecích linek při zvýšení tepelného příkonu K3-4, odstavováním mlecích linek při snížení tepelného příkonu K3-5 a start záložní mlecí linky při výpadku některé z provozovaných mlecích linek autonomní ochranou K3-6. Schéma součinnosti algoritmu je uvedeno v příloze 1.

Jednotlivé algoritmy jsem ponechal v původní formě, ale v hlavním funkčním subalgoritmu K3-4 jsem provedl zásadní změny, které uvedu v následujícím popisu jednotlivých subalgoritmů.

3.1 Jednotlivé subalgoritmy

K3-1 registruje stavy tlačítek start, stop a ovládá příslušné signálky podle stavu rozhodovacích podmínek. Po odstavení mlecích linek odstavuje tento program mazutové hořáky. Podle některých vnitřních proměnných blokuje činnost příslušných subalgoritmů a konečně prověřuje vstupní rozhodovací podmínky subalgoritmu a podle jejich stavu je vyvolává z vnější paměti pomocí aktivace logických řídicích proměnných XX 3 až XX 6.

K3-2 je pomocný program, který ruší veškeré povely start-stop v dalším cyklu. Tím je délka těchto signálů vymezena na dobu 10 sek. K3-3 odstavuje mlecí linky do studeného stavu v podstatě naráz na povel tlačítkem stop za předpokladu,

že výkon bloku dosáhne hodnoty 100 MW.

K3-4 lze rozdělit na tři základní části. V první části po prověření vstupních podmínek lze zapnout mazutové hořáky / pokud mazuty z jakýchkoliv důvodů nelze najet je obsluha podáno hlášení " mazuty nelze najet " a program se ukončí a zablokuje. Obsluha pak buď závadu odstraní, nebo nastartuje mazuty ručně a chce-li, aby program spuštění pokračoval automaticky deblokuje program K3-4 opětovným stisknutím tlačítka start / a pokud trvá požadavek zvýšení tepelného příkonu, je prověřován stav motoru jednotlivých mlecích linek, eventuálně při najetí po havarii teploty za třidičem. Linky s nezapnutými motory, eventuálně s vyšší teplotou za třidičem / 150 - 180 °C/ jsou ve stanoveném pořadí označeny C 11 až C 16. Počet zapnutých linek, zvlášť pravé a levé strany spalovací komory, je zapsán do číselných pamětí těchto stran. Pokud C 11 - C 16 je rovno 0, je program ukončen. V první části je rovněž dotaz na obsluhu, zda-li chce provádět start automatů linek automaticky s určitou čekací dobou na odpověď.

Pokud se provádí automatické najiždění, uvede se v činnost druhá část. Pomocí proměnné C 30 / rozdíl mezi levou a pravou stranou / se zjistí, zda je vhodnější zapnout mlecí linku na levé nebo pravé straně anebo na tom nezáleží. Pokud je výhodné zapnout mlecí linku na levé straně / P 7 je menší než P 8 / přiřadí se pomocným proměnným U-2. U 4, U 6 příslušné hodnoty C 12, C 14, C 16 a C 12, C 14, C 16 se nuluje. Pokud je výhodnější zapnout linku na pravé straně, / P 7 je větší než P 8 / přiřadí se pomocným proměnným U 1, U 3, U 5 příslušné hodnoty C 11, 13, 15 a C 11, 13, 15 se vynuluje.

Ve třetí části použijí C 6 jako index u proměnných C 11 - C 16,

/ C 1, C 6 /. Zjistím, je-li proměnná C 1, C 6 rovná jedné to znamená, že příslušná ML je vypnuta. Zjistím, opět pomocí C 6, o kterou ML jde, přiřadím hodnoty vybrané ML k obecným hodnotám, které se použijí v části algoritmu zapínání mlecích linek a přejdu na část algoritmu zapínání ML. Je-li proměnná C 6 než 6 dojde k dotazu zda-li je požadováno zvýšení výkonu. V kladném případě se obsluze nahlásí " nelze zvýšit výkon " a algoritmus je ukončen v záporném případě je algoritmus ukončen.

Část algoritmu zapínání ML začíná prověřením vstupních podmínek, jež byli přiřazeny k obecným proměnným a dojde ke startu příslušné mlecí linky. Pokud podmínky startu splněny nejsou provede se hlášení " podmínky startu nesplněny", je hlídán časový limit vybrané ML pomocí časové proměnné A. Pokud tato nenabude hodnotu jedné a dojde ke spuštění, je ML zapsána do příslušné číselné paměti. Vybrání paměti je provedeno pomocí pomocné proměnné N, která označuje číslo mlecí linky.

V další části je učiněn dotaz trvá-li požadavek na zvýšení příkonu tepla. Pokud ano zkontroluje se je-li některá z proměnných C 11 - C 16 rovna 1. V záporném případě dochází podle stavu rozdílu číselných pamětí levé a pravé strany k přiřazení hodnot U 2, U 4, U 6 proměnným C 12, C 14, C 16 / P 7 meší než P 8 /, U 1, U 3, U 5 proměnným C 11, C 13, C 15 / P 7 je větší než P 8 / a opět se kontroluje, zda-li je některá z proměnných C 11 - C 16 rovná 1. Provádí se kontrola na vhodnost konfigurace. Pokud by vznikla nepřípustná konfigurace je obsluha informována hlášením " nepřípustná konfigurace " a obsluha, po odstranění závad, opět spustí program. Pokud je konfigurace nanejvýš nevhodná provádí se přiřazení hodnot a zapínání ML.

Netrvá-li požadavek na zvýšení příkonu tepla K 3, 4 se ukončí. Trvá-li požadavek na zvýšení příkonu tepla a některá

z proměnných C 11 - C 16 je rovná jedné, je po uplynutí limitu, pokud došlo k zapnutí ML a jinak ihned, vybrána další linka.

K 3 - 5 je řešen obdobným způsobem jako K 3 - 4. Řídící proměna RPK 3 nabývá při jeho přerušení hodnoty 2.

K 3 - 6 startuje záložná mlecí linka, jakmile některá z mlecích linek byla odstavena svoji autonomní ochranou. Obdobně jako K3-4 sestává ze tří základních částí. V první části zapíná krátkodobě mazutové hořáky, ve druhé části prověřuje konfiguraci pro zapínanou záložní linku a ve třetí části tuto linku, za předpokladu splněných podmínek startuje. Pokud došlo v K3-6 k zapnutí mazutů, pak jsou ve třetí části po startu záložní linky vypnuty. Po zapojení mlecí linky se algoritmus sám zablokuje a jakmile přestane autonomní ochrana působit, může obsluha K3-6 deblokovat stisknutím tlačítka " start ".

3. 3 Rozhodovací tabulky jako vhodný nástroj zápisu logického řízení.

3. 3. 1 Vývojové diagramy a rozhodovací tabulky

Při řešení zápisu problémů, či rozhodovacího modelu je třeba splnit některé podmínky, které vyžaduje příslušné řešení. Je to předně:

- přesné a jednoznačné vyjádření složitého logického problému,
- uspořádání do přehledné formy, umožňující snadnou orientaci a manipulaci v dynamickém prostředí,
- snadnou realizaci změn modelu,
- jednoduchou a čitelnou úpravu, srozumitelnou širokému okruhu spolupracovníků.

Některé z těchto požadavků dokázala splnit technika, která byla vyvinuta v souvislosti se zaváděním automatizace. I zde byl primární požadavek přesné a jednoznačné vyjádření složitého logického problému ve formě umožňující snadnou realizaci změn.

Protože však nebyl kladen důraz na uspořádání do přehledné formy, umožňující snadnou manipulaci přímo na pracovišti, nepronikla ve větší míře tato nová technika - vývojové diagramy do organizační praxe.

Vývojové diagramy se však velmi rozšířily mezi analytiky a programátory, protože dokáží poměrně dobře plnit i další skupinu požadavků, které byly definovány v jednotlivých fázích automatizovaného zpracování problémů.

Specifické vlastnosti počítačů umožnily zabývat se problémy mnohem složitějšími a náročnějšími, než které bylo dříve možno řešit jinými technikami. Formální zápis tohoto rozsáhlého problému musel být tedy nejen uzpůsoben požadavku vyjádřit značně větší počet logických vazeb a podstatně složitější algoritmus rozhodování, ale musel i v podmínkách týmové práce, kterou si rozsah řešení problematiky vynutil, být i jednoznačným dorozumívacím jazykem mezi pracovníky různých profesních specializací, kteří se na budování automatizovaného systému podílejí od řídicích pracovníků a analytiků přes techniky a programátory až po operátory i různé pomocné profese.

Jaké jsou tyto další požadavky, které vyplynuly z jednotlivých fází budování automatizovaného systému? Jde především o:

- vytvoření dokonalého nástroje jednoznačné formulace problému, by zároveň umožnil přesné a detailní vystižení algoritmu požadované v další fázi automatizovaného zpracování v programování;
- splnění požadavků přehledné dokumentace problému, nutné pro usnadnění komunikace, mezi jednotlivými členy týmu i pracovníky, kteří budou v budoucnosti v případě změny zajišťovat aktualizaci zpracovaného problému;

- vytvoření takového formálního zápisu algoritmu problému, který by byl maximálně přenositelný, tedy nezávislý na používaném typu počítače a jeho jazyku.

Metoda vývojových diagramů dokáže tyto požadavky splnit téměř beze zbytku. Během doby se však ukázaly některé slabiny této metody. Především; v rozsáhlých vývojových diagramech s komplikovanou logikou je značně obtížná orientace. Analytik nemá jistotu, zda uvažil všechny kombinace, ke kterým může dojít.

Vývojové diagramy jsou jinak vhodné především v oblasti analýzy a algoritmizace, ale umožňují do větší míry respektovat požadavky návazných etap budování ASŘ, protože zůstávají jen na úrovni jednotného dorozumivacího jazyka automatizátorů. Na jedné straně jsou poměrně málo srozumitelné organizačním pracovníkům a řídicím kádrům, které s nimi v souvislosti s budováním automatizovaných systémů řízení přicházejí stále více do styku na druhé straně toto vlastní vyjádření algoritmu nerespektuje dokonale ani požadavky další fáze automatizovaného zpracování

- etapy programování.

Koncem padesátých let byly vyvinuty rozhodovací tabulky, které nejen důsledně splňují všechny dříve definované požadavky a umožňují tedy, narozdíl od vývojových diagramů, velmi přehledně vyjádřit i nejsložitější logické vztahy a vazby. Velmi dobře splňují i kryterium využitelnosti takto zapsaného rozhodovacího modelu přímo v řídicí a organizační práci, protože tento logický zápis nevyžaduje speciální znalosti automatizačních technik. Navíc plní rozhodovací tabulky i další skupinu požadavků návazných etap budování ASŘ. V etapě programování je to požadavek:

- umožnit zápisem algoritmu minimalizaci rozsahu programu

využívané paměti;

- zajistit dostatečnou rychlost překladu i rychlost zpracování přeloženého programu;
- uspořádání zápisu algoritmu do formy, která zajistí maximální využití možností vyšších programovacích jazyků;
- zjednodušení dokumentace systému;

V etapě ladění:

- usnadnění přípravy testovacích dat;
- usnadnění dohledání chyb, tím zkrácení ladění a zavedení systému.

Plnění všech požadavků definovaných v této stati znamená, že uplatnění rozhodovacích tabulek je možné v nejrozšířenějších oblastech a že tato technika může vytvořit nejrozšířenější a nejefektivnější dorozumivací nástroj, který jistě přesahuje důležitost všech zmíněných předcházejících technik.

3. 3. 2 Strukturní prvky a druhy RT

Dvourozměrné tabulkově zspořádané informace jsou v praxi běžně využívány jako nástroj umožňující vyjádřit složité vazby mezi dvěma množinami proměnných. Tato technika však dává pouze statický pohled na problém. Je možné tímto způsobem vyjádřit pouze vzájemné vztahy mezi hodnotami nezávisle a závisle proměnných na základě předem definovaných pravidel, aniž bychom byli schopni z takového běžného tabulkového zápisu vyčíst rozhodnutí o tom, jaký sled činností má z těchto různých kombinací stavů nezávisle proměnných být odvozen.

Na rozdíl od tabulkového zápisu, používaného v běžné praxi i pravdivostných tabulek matematické logiky, dávají tuto možnost RT.

Aby bylo možno dodržet exaktnost zápisu i jednoznačnost interpretace rozhodovacího modelu zapsaného v RT, musí se

dodržet jednotná forma i struktura a tabulka musí mít určité náležitosti. Pro názornost zde uvedu schéma RT, které platí obecně pro všechny RT.

1.kvadrant seznam podmínek	3.kvadrant kombinace podmínek
2. kvadrant seznam činností	4.kvadrant kombinace činností

Jak vyplývá ze schématu, tvoří jednotlivé kvadranty čtyři základní části RT, které musí být od sebe odděleny a mají své stálé místo. RT tedy definují všechny podmínky, to je nezbytné předpoklady, které musí předcházet, mají-li následovat činnosti / 3 kvadrant /. A ty musí následovat jestliže předcházejí definované podmínky / 1 kvadrant /. Pravá strana tabulky pak tvoří různé kombinace těchto podmínek a z nich vyplývajících činností, tedy jednotlivá logická pravidla. Podmínky musí být stanoveny tak, že kdykoliv nastane potřeba rozhodnout o určitém případě, může se použít pouze jediné pravidlo.

Toto jediné pravidlo se může použít teprve tehdy, jestliže jsou uspokojeny všechny podmínky platné pro toto pravidlo a pak musí následovat všechny činnosti vyznačené ve spodní polovině pravidla, a to v předepsaném pořadí.

Pravidlo je tedy jeden sloupec v pravé polovině tabulky, který má dvě části: určitou podmínku / či určitou kombinaci podmínek /, kterou následuje určitá činnost / nebo určitý sled činností /.

Existují tři druhy RT

- 1. RT s vymezenou volbou /sjednoduchými vstupy/.

Tyto RT se vyznačují tím, že v levé části, t.j. v polovině specifikaci je v 1. kvadrantu zachycen detailní rozpis podmínek a ve 2. kvadrantu detailní rozpis činností. Pak mohou být kombinace stavu podmínek /3.kvadrant/ vytvářeny pouze symboly "A" = ano, "N" = ne a "-" nebo prázdné místo = není rozhodující. Ve 4. kvadrantu - výběr činností se používají pouze symboly "X" = tato činnost nastane nebo "-" či prázdné místo = tato činnost nenastane. Činnosti se musí provést v tom pořadí, v jakém jsou zapsány.

Exaktnost tohoto zápisu i vysoký stupeň formalizace vytváří sice RT většího rozsahu než dále uvedené druhy, ale tento zápis je velice vhodný pro využití ve vazbě na počítač. V řídicí a organizační praxi však skýtají větší možnost RT jiného typu.

- 2. RT s rozepsanou volbou /rozšířenými vstupy/.

Zásadní rozdíl mezi jednoduchými a rozšířenými rozhodovacími tabulkami je ve způsobu zápisu podmínek a jejich vstupů. Zatím co v prvním případě se v prvním kvadrantu rozhodovací tabulky uvádí kompletní formulace podmínky, kterou lze zodpovědět pouze ANO či NE, vzniká v rozšířených rozhodovacích tabulkách kompletní znění podmínky až teprve sloučením části zapsané v seznamu podmínek a vstupu této podmínky v některých z pravidel. Hovoříme proto o tom, že u rozšířených rozhodovacích tabulek představují jednotlivé vstupy podmínek komplementy části podmínek uvedených v seznamu podmínek. Jinými slovy, každá podmínka z prvního kvadrantu má tolik formulací, kolik rozdílných vstupů obsahuje ve všech pravidlech rozhodovací tabulky.

- 3. RT kombinované /se smíšenými vstupy/.

Jak již sám název napovídá, jsou přípustné oba dosud pop-

popsané způsoby zápisu podmínek a jejich vstupů a činností a jejich vstupů. Z hlediska praxe jsou smíšené tabulky nejvýdělejší, přesto se jich využívá v různých oblastech nerovnoměrně. Nejvíce se uplatnily jako všeobecné a analytické tabulky, ale takřka se jich nepoužívá jako programových tabulek. Navíc uživatelé mimo počítačovou oblast a systémoví analytici dávají přednost jednoduchým rozhodovacím tabulkám pro jejich přehlednost a jednoznačnost při odhalování možných logických chyb.

- Zvláštním druhem RT jsou RT činnosti kde chybí podmínková část. Z důvodu dodržení jednotné formalizace zápisu, vyjádří se prostý sled činností nezávislých na podmínkách rovněž rozhodovací tabulkou. Jde o analogii vývojového diagramu vyjádřeného pouze operačními bločky. Tímto způsobem se řeší např. činnosti, které se vícekrát opakují.

3. 3. 3 Zmenšování rozsahu RT.

Počet pravidel RT je závislý na počtu podmínek zapsaných v RT a na počtu vztahů, který může každá podmínka nabýt. Je tedy počet kombinací, na rozdíl od vývojových diagramů, předem zjistitelný. U RT s vymezenou volbou je počet kombinací 2^n , kde n je počet podmínek. Ne všechny pravidla se však využívají a proto se snažíme o co nejmenší rozsah RT, při zachování správnosti RT. Jednou ze základních možností je vyloučení ne-logických kombinací, to jest takových, které nemohou z hlediska řešeného problému nastat. Další možností je slučování pravidel, a to tak, že dvě pravidla se mohou sloučit do jediného, jestliže se liší pouze v jedné podmínce a jestliže u obou následuje stejná činnost /stejný sled činností/. Toto prázdné místo říká, že pro tuto kombinaci může být stav A i N, tedy jsou zde obsaženy obě původní pravidla.

Třetí cesta spočívá v použití pravidla JINAK. Tento druh

zápisu do RT se používá tehdy, kdy RT s větším počtem podmínek je pro hledaná rozhodnutí nutný jen malý počet pravidel a všechna ostatní pravidla mají společné rozhodnutí /postup/. Pak se rozepíše těchto několik významných kombinací do jednotlivých pravidel a ostatní se shrnou do jediného pravidla JINAK. Pravidlo JINAK je v RT poslední a u žádné z podmínek nesmí být v tomto pravidle uveden významný stav.

3. 3. 4 Dělení a spojování RT.

Rozsáhlé problémy obvykle není možné vyjádřit jedinou RT. Buď je počet podmínek tak velký, že při kvadratické závislosti počtu pravidel na počtu podmínek /u RT s vymezenou volbou/ by zápis do RT byl nerealizovatelný /nebo alespoň nepřehledný/, nebo zápis tak rozsáhlého modelu do jediné RT je vyloučen samotným problémem. Toto nastává tehdy, jestliže po nějakém testu skupiny podmínek musí proběhnout řada činností a pak se testují další podmínky ovlivňující další volbu postupu. Samotná logika tedy rozděluje úlohu do několika dílčích rozhodovacích modelů.

RT samozřejmě mají možnost vyjádřit tyto vazby mezi jednotlivými dílčími modely, a to formou jak volného přechodu z jedné RT na druhou /či na sebe sama/, tak formou odskoku na jinou RT s návratem na původní místo, při opakovaném využívání dílčího algoritmu.

Technicky se tento přechod realizuje příkazy:

Přejdi na název RT

GOTO název RT

NÁVRAT

Tyto příkazy se zapisují jako poslední v každém pravidle RT z něhož se má odskok případně návrat uskutečnit.

3. 3. 5 Převod RT do programu.

Některé z podstatných výhod metod RT proti jakékoliv jiné metodě použité pro zápis algoritmizace problému určeného k převodu do programu jsou např.:

- uzpůsobení zápisu do tvaru, který maximálně umožňuje využít výhod vyššího programovacího jazyka především ve směru spojování podmínek, které se testují najednou do jediného příkazu;
- zápis třetího kvadrantu, kde jsou definovány všechny možné kombinace umožňuje aplikaci některého optimalizačního postupu a tím dosažení zrychlení průběhu generovaného programu, případně zmenšení nároku na potřebnou paměť;
- přehledné uspořádání 4 kvadrantu umožňuje pro opakovaně se vyskytující sled činností minimalizovat i kód generovaný pro část činností užitím společných bloků.

Jednou z nejjednodušších metod převodu RT do programu je sestavení programu po pravidlech. Zaměřuje se především na podchycení výhod vyšších programovacích jazyků: slučovat do jednoho příkazu více podmínek, případně zlepšit průběh programu podle frekvence výskytu jednotlivých kombinací /pravidel/.

Obečné dispozice:

- je-li uvedena frekvence výskytu jednotlivých pravidel, je možné sestavit "optimální" program tak, že pravidlo s největší frekvencí se zapisuje jako první,
- není-li uvedena frekvence výskytu, může se jako první zapisovat pravidlo s největším počtem prázdných míst /komplexní pravidlo/, u kterého je teoreticky pravděpodobnost nejčastějšího výskytu, pak pravidla s menším počtem prázdných míst, nakonec pravidla jednoduchá,
- při stejné pravděpodobnosti výskytu jednotlivých pravidel se obvykle postupuje při sestavení programu zleva doprava.

Druhou metodou, kterou jsem ve své práci použil je sestavování programu s použitím zjednodušeného Pollackova algoritmu. Jde v podstatě o vyjádření 3. kvadrantu RT rozhodovacího stromem, který by měl minimální počet uzlů. Má-li totiž RT nezávislé podmínky a alespoň i komplexní pravidlo, je počet uzlů rozhodovacího stromu / a tím i programem definovaných podmínkových příkazů IF../ různý podle toho, v jakém pořadí se provede test podmínek.

Pro určení pořadí rozkladu podmínek je rozhodující počet prázdných vstupů každé podmínky. Při rozkladu se jako první použije podmínka, která má nejnižší počet prázdných vstupů /tedy nejvíce explicitně vyjádřených stavů A, N/ a jako poslední ta, která má nejvíce prázdných vstupů.

Jestliže více podmínek má podle tohoto kritéria stejnou hodnotu /Pi/, použije se mezi nimi ještě pomocné výběrové kritérium tzv. hodnota Di. U dvou nebo více podmínek se stejným počtem prázdných vstupů se pro rozklad dříve použije ta, kde rozdíl mezi počtem kladných stavů podmínek a jejich záporných stavů je větší.

RT se rozkládá podle určeného pořadí podmínkových testů vždy na dvě podtabulky, které jsou zmenšené o podmínku, podle níž se právě provádí test /rozklad/:

- první tabulka obsahuje všechny zbývající podmínky a pravidla, u nichž má výchozí podmínka stav "A" a stav "-";
- druhá podtabulka všechny zbývající podmínky a pravidla, u nichž má podmínka stav "N" a "-".

Pro ruční sestavení programu z použitím stromového rozkladu a Pollackova optimalizačního postupu byl celý algoritmus značně zjednodušen a je pro tento účel postačující.

Třetí použitá metoda je pomocí Veinottova algoritmu vypočteného kroku. Tato metoda využívá z typických vlastností RT, a to té, že v závislosti na podmínkách, které do RT vstupují je rozepsán komplexní rozhodovací model, t.j. podle kombinace stavů podmínek je každé kombinaci přiřazen určitý postup. Algoritmus z každé kombinace podmínek odvodí jedinou adresu, na které je uložen postup předepsaný pro tuto kombinaci.

Jestliže se každé podmínce přiřadí váha, která je mocninou 2 /od 2^0 až 2^n , kde n je počet podmínek/, pak se může zjistit celková váha pravidla tak, že kladné stavy tohoto pravidla, násobené vahou podmínky, u které k tomuto kladnému stavu došlo, se sečtou a výsledná číslice může tvořit návěští odstavce, kde jsou umístěny činnosti tohoto pravidla. Váha pravidel stoupá po 1 od nuly do n . Připočteme-li k této váze pravidla 1, dostaneme pořadí, na kterém se najde hledaná kombinace. Tato čísla pořadí může pak být součástí příkazu pro vypočtený skok.

3. 3. 6 Automatizovaný převod RT do počítače.

Píšu-li o využívání RT v budování ASŘ, bylo by chybou nezmínit se alespoň krátce o automatizovaném převodu RT do programu. Pro plné využívání výhod RT jsou softwarové převodníky takřka nezbytné, proto by bylo vhodné se v dalším vývoji využívání této metody zaměřit zvláště na tuto stránku.

3. 4 Popis vyjádření algoritmu K3. 4 pomocí RT.

Systém RT Algoritmu K3. 4 jsem rozdělil z důvodu značné rozsáhlosti úlohy do 6. skupiny RT, /příl.3/, které jsou součástí zásad kapitoly 3.3.4. Protože však jednotlivé skupiny byly pro zpracování do jedné RT příliš rozsáhlé, každou jsem opět rozdělil dle zásad kapitoly 3.3.4 do několika RT.

Skupina "KONTROLA A NASTAVENÍ HODNOT" - RT 1, RT 2, RT 3

vyberou podle zvolené ML příslušné hodnoty proměnných, potřebných při spouštění ML a přiřadí je obecným proměnným. Zároveň se vynuluje prvek vektoru indikace vypnuté ML. Tato skupina je rozdělena do tří jednoduchých RT smíšeného typu. Po vybrání a přiřazení hodnot se přechází na skupinu RT "ZAPÍNÁNÍ ML".

Skupina "ZAPÍNÁNÍ ML" je, z důvodu přehlednosti, rozdělena do tří skupin RT. V první RT / příl. 3 RT 8.1/ se, podle hodnoty příslušných proměnných, zapíná motor vybrané ML. V případě, že ML nelze najet, přiřadí se příslušné logické proměnné indikující klášení "1". V druhé RT / obr. 12 RT 8. 2/ se kontroluje, je-li požadavek na zvýšení tepla a je-li v pořadníku nějaká volná linka. Dále prochází limit a v případě splnění se přiřadí příslušné logické proměnné "1". Třetí tabulka RT 8. 3 / obr. 12/ prověří zda je nějaká ML volná a v kladném případě se přechází na skupinu tabulek "KONFIGURACE". V záporném případě se přechází na RT 20H.

Mimo popsaných RT, jsem ve zpracování algoritmu K3.4 požil dvě RT činnosti / kapitola 3.3.2/. Jde o RT 20H /příl. 3/, která provede sekvencí činností, jež se několikrát opakuje na různých místech programu a z ní se přechází na počátek programu. Druhá tabulka RT C9 /příl. 3/ se používá vždy, vyskytuje-li se v algoritmu prodleva. V tom případě tabulka znázorňuje deseti vteřinový limit čekání a pak se vrací do místa algoritmu ve kterém prodleva probíhá.

3. 5 Srovnání metody RT s metodou vývojových diagramů.

Výhody jež mají RT oproti vývojovým diagramům jsou tyto:

- zápis v RT je úspornější,
- je možná kontrola na úplnost,
- je přehlednější, obzvlášť při větších úlohách, kdy vývojový diagram musí být rozepsán na více stran,

- dává lepší předpoklady pro uplatnění progresivních přístupů v oblasti projektování a programování ASŘ, zejména modulárního, normovaného strukturálního projektování a programování,
 - je čitelnější a pro pracovníky z oblasti řízení, organizace a administrativy, se kterými nutně musí systémový pracovník při budování ASŘ komunikovat,
 - lépe vyhovuje potřebám promítání změn do rozsáhlejších úloh. Výměna jedné RT je snazší než oprava úseku vývojového diagramu, a dále je možno říci, že RT je:
 - vhodnější podklad pro programování než vývojový diagram, protože dává lepší možnost využití výhod vyšších programovacích jazyků, zejména slučování více podmínek do jediného příkazu a protože lze v ní uplatnit optimalizační metody, kterými se zvýší efektivnost programu připraveného z RT nebo sníží nároky na vnitřní paměť počítače,
 - výborný podklad pro přípravu testovacích dat, protože na rozdíl od vývojových diagramů lze pro testování úlohy připravit podle RT kompletní soubor dat ověřující všechny možné kombinace,
 - využitelné i jako přímý vstup do počítače, protože pomocí převodníků lze z RT přímo generovat program počítače.
- Obecně je tedy možno konstatovat, že RT jsou vhodnější nástroj v budování ASŘ než vývojové diagramy.

4 Popis programu

Program má strukturu podle RT oproti nim nastala některá zjednodušení, která umožňuje programovací jazyk RPP Fortran a některé názvy veličin byly z důvodu gramatiky jazyka pozměněny.

4. 1 Převod RT do programovacího jazyka

RT 1 jsem převedl do programovacího jazyka metodou zjednodušeného Pallackova algoritmu. Jako příklad uvádím tuto metodu

/ příl. 3/ je rozdělena do tří částí. V první /příl. 3 1-RT 1/ části je 1 RT, ve které se v závislosti na stupních hodnotách provede nulování hlášení, vektorů P7 a P8 a vektorů indikace zapnutí jednotlivých ML.

Ve druhé části se provede nastavení vektorů P7, P8 a C11-16 při normálním provozu. Tato část je hlavně z důvodu jednoduchosti a přehlednosti rozdělena do tří RT. /příl. 3- RT 2, 1, RT 2.2, RT 2.3/.

Třetí část je rozdělena stejně jako část druhá a provádí totéž avšak po havárii, tedy v závislosti teploty za třidičem / příl 3 - RT 3.1, RT 3.2, RT 3.3/.

Skupina "ZAPÍNÁNÍ MAZUTOVÝCH HOŘÁKŮ" je rozdělena do 4.RT. V první RT /příl. 1 RT 4.1/ se kontroluje, zda je třeba zapnout MH. Druhá tabulka / příl. 3 RT 4.2/ zapíná MH a v případě , že Nelze MH zapnout, přechází na tabulku 2OH. Třetí a čtvrtá tabulka /příl. 3 RT 4.3, RT 4.4/ podle přání obsluhy buď stavějí celý algoritmus, nebo pokračují v automatickém provozu.

Skupina "ROZHODOVÁNÍ" /příl. 3 RT 5.1, RT 5.2/ je rozdělena na dodvou jednoduchých RT, které řeší, podle hodnoty proměnné C 30, zda je výhodné zapínat ML na levé či pravé straně. Příslušná ML se vybere a přechází se na tabulku "PŘEPÍNAČ".

Skupina "KONFIGURACE" má tři části / příl. 3 a obr. 9/. V první je jedna RT smíšeného typu, která prověřuje zda kontrolovat z hlediska vhodnosti konfigurace levou či pravou stranu /příl. 3 RT 6.1/. Druhá a třetí část jsou paralelní. Obě jsou složeny, z důvodu přehlednosti, ze dvou tabulek, přičemž druhá část /příl. 3 RT 6.2 a RT 6.3/ řeší konfiguraci levé strany a třetí /příl. 3 RT 6.4 - RT 6.5/ pravé strany.

Tabulky "PŘEPÍNAČ" /příl. 3 RT 7.1, RT 7.2 a RT 7.3/ vy-

u RT 4.1, která je kratší a pro jednoduchost této metody, jako demonstrace plně postačující.

RT2.1, RT2.2, RT2.3, RT3.1, RT3.2 a RT3.3 mají shodnou strukturu, a proto jsem této vlastnosti shodně použil v programu, kde jsem tento problém řešil metodou podprogramu.

Po zvážení všech možností převodu RT do programu jsem u podprogramu znázorňujícího RT 2 použil pro převod metodu Veinotova algoritmu vypočteného skoku. Pro demonstraci uvádím převod tabulky RT 2.1.S OBECNÝMI PARAMETRY

NÁVĚSTÍ	29	27	26	28
A = 1	A	A	N	N
B = 1	A	N	N	A
P7 = 1+ P7	X	X		
P8 = 1+ P8	X			X
RETURN	X	X	X	X
C = 1			X	X
D = 1		X	X	
VÁHA	3	1	0	2

```

IF / A - 1 / ,2,1,1;
1 I1=1;
GOTO5;
2 I1=0;
5 IF /B-1/,4,3,3;
3 I2=1;
GOTO6;
4 I2=0;
6 J=1+I1+2xI2;
GOTO /26,27,28,29/,J;
26C=1;
D=1;
RETURN;
27 P7=1+P7;
D=1
RETURN;
28 P8=1+P8;
C=1;
RETURN;
29 P7= 1+P7;
P8=1+P8;
RETURN;

```

RT 2.1, RT 2.3 , RT 2.2 řeší podprogram KS. Skupinu RT 3.1, RT 3.1. 2, RT 3.1. 3 řeší podprogram HAVAR. Tento je sestaven metodou zjednodušeného Pollackova algoritmu.

RT 4.2 až RT 4.4, 5.1,5.2,6.1 jsem převedl metodou po pravidlech, protože to dle kapitoly 3.3.5 bylo nejvýhodnější. Pro představu o jednoduchosti této metody uvádím přepis druhého pravidla z tabulky 5.1

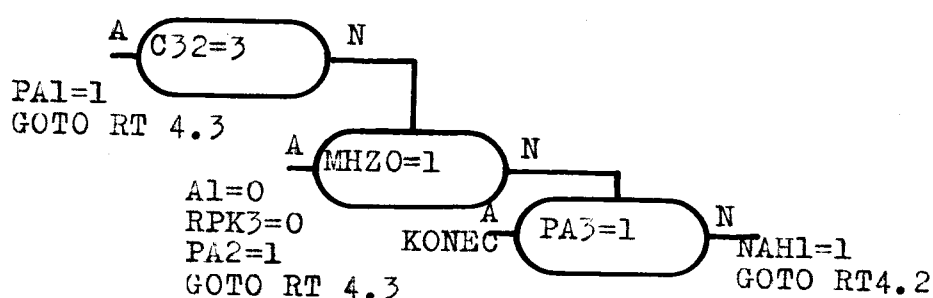
C 30 : 0	=	=
C 12,14,16, = 0		
C 6 = 1		X
GOTO TAB C2		X

IF/P7-P8/,44,46,47

46 C6 = 1

GOTO TAB C2

Tabulku RT 4.1 jsem převedl do programu zjednodušeným Pollackovým algoritmem. Pro demonstraci uvádím vývojový diagram sestavený Pollackovou metodou z této tabulky.



Tabulky RT 7.1,7.2,7.3, jsem převedl po pravidlech. V programu jsem použil index N, který registruje číslo linky. Je využit v hlášeních. Tabulky 8.1, 8.2, 8.3, jsem převedl po pravidlech.

Volbou různých metod přepisů jsem jednak chtěl optimalizovat program a jednak si v praxi vyzkoušet metody přepisu RT, uváděné v literatuře.

4. 2 Popis uživatelského programu.

Program byl sestaven z RT podle zadání EGU, proto jsem ponechal všechny proměnné stejně označeny, pokud to bylo možné z hlediska programovacího jazyka RPP fortran. C6 jsem nahradil proměnnou K, protože ji používám jako index u vektoru C1 /K/.

Na začátku programu jsem závislosti na hodnotě RPK 3 vynuloval proměnné indikující hlášení / RPK 3 = 0/. Je-li RPK 3 = 1 provede se pomocí přepínačového příkazu skoku odskok na

místo přerušení. Určení místa se vyhledává proměnnou L.

Dále se provádí kontrola a nastavení hodnot jednotlivých ML. Toto je provedeno pod programem KS, v případě, že jde o najetí po havarii pod programem "HAVAR". Po tomto úkonu se provádí najetí na MH / v případě, že je to nutné - C 32 je menší než 3/. Nedojde-li k tomu /MHZO = 0/ hlásí se tato skutečnost na tiskárně tiskem "MH NELZE NAJET". Dojde-li k najetí provádí se na tiskárně "LIMIT SPLNĚN". Po kontrole, jeli stále požadavek na zvýšení příkonu /QZL 4=1/ a zda je ještě nějaký motor ML nezapnut /L10=0/ dochází ke startu ML automaticky. To je též hlášeno na tiskárně "START ML AUTOMATICKY". Není-li žádný motor volný, provede se hlášení "NELZE ZVÝŠIT VÝKON" a ukončí se výpočet.

Chce-li se automaticky provoz /A18=1/ na tiskárně se tiskne "AUTOMATICKY" a dále je řízení zapínání a vypínání jednotlivých ML automatické. V závislosti na rozdílu zapnutých ML na levé a pravé straně se provede: je-li P7 větší než P8 a HL 4 není větší jak 1 pomocným proměnným U1 U3 a U5 se přiřadí číselné hodnoty C 11, C13 a C15 a ty se vynulují. Je-li P8 větší jak P7, a HL4 není větší jak 1 pomocným proměnným U2 U4 a U6 se přiřadí číselné hodnoty C12, C14, a C16 a tyto se vynulují. Je-li P7=P8 nebo HL4 je větší než 1, přechází se na výběr ML, která se bude zapínat.

Číslo ML, jež se bude zapínat se určí dle hodnoty C1/K/. Je-li tato rovná 1, skáče se na přiřazení hodnot určené ML obecným hodnotám subalgoritmu zapínání ML. Není-li žádná z hodnot určené ML obecným hodnotám subalgoritmu zapínání ML. Není-li žádná z hodnot C1 /K=1/, buď se všechny ML už zaply, nebo ty co se nezaply nejdou najet a je požadavek zvýšení výkonu dochází k tisku "NELZE ZVÝŠIT VÝKON". V opačném případě se

vybere ML a opět se skáče na přeřazování hodnot vybrané ML. To se uskuteční pomocí hodnoty K.

Po přiřazení se provádí zapínání ML. Není-li možné z důvodu nesplněných podmínek vybranou ML zapnout, tiskne se "NESPLNĚNÝ PODMÍNKY" ML N.N značí číslo ML, u nichž k tomuto došlo a skáče se na dotaz je-li požadavek zvýšení výkonu. Je-li tomu tak určí se pomocí proměnné $C17/součet\ C1/K$, zda je možno zapínat další linku a přechází se buď na výběr nové linky anebo na kontrolu konfigurace. Je-li konfigurace nepřipustná, dojde k tisku "NEPŘÍPUSTNÁ KONFIGURACE". Jinak dochází opět k výběru ML.

Jsou-li podmínky ML splněny dochází k jejich najetí. Nedojde-li k tomu v určeném limitu, tiskne se "LIMIT PŘEKROČEN U MLN" a výpočet se ukončí. Dojde-li k najetí ve splněném limitu tiskne se "MLN ZAPNUTA".

Dále se kontroluje je-li požadavek zvýšení výkonu. V záporném případě se výpočet ukončí, v kladném se čeká na uplynutí limitu po najetí ML. To se ohlásí tiskem "LIMIT PO NAJETÍ SPLNĚN". V dalším cyklu se opět provádí najíždění v automatickém provozu /návestí 38/.

Program tedy provádí všechny činnosti, které souvisejí s najetím ML v normálním provozu i po havarii. Hlášení obsluhy jsou prováděna na tiskárně. Po každém výpočtu je program ukončen. Pro praktické použití je třeba, aby bylo umožněno kontrolovat všechny hodnoty a případně provádět určené činnosti v určitém intervalu /10 sec./ Tento požadavek však neumožňuje počítač RPP 16 - S při použití jazyka RPP Fortran.

Časové prodlevy jsou reprezentovány logickou proměnnou. Je-li prodleva vyčerpána nastaví se hodnota 1.

Zadání konkrétních dat z technologického procesu jsem neřešil, protože bylo mimo rozsah práce i mimo možnosti, které jsem měl při sestavování a ladění programu. Vzhledem k tomu, že jsem však dodržel všechny požadavky zadavatele a ponechal shodné označení vstupů, výstupů i vnitřních proměnných, lze program po výše uvedených úpravách použít. Program v příl. 4.

4. 3 Popis simulačního programu.

Simulační program vychází z programu uživatelského. Zadávání vstupních hodnot je pro časté změny řešeno v PS. Pro kontrolu se vstupní hodnoty hned po zadání na tiskárně ve stejném pořadí, v jakém byly zadávány na PS. Dále se provádí výpočet podle uživatelského programu.

Časové prodlevy jsem simuloval tak, že jsem při najetí na jakoukoliv z nich přerušil výpočet s dotazem na PS - je-li příslušný limit splněn, nebo proběhla-li činnost v určeném limitu. Z PS jsem potom zadával potřebné hodnoty. Při splnění časového limitu "1", při trvání limitu "0". Dále jsem ještě při přerušení zadával hodnoty z technologie - MH a ML ZAPNUTY popř. VYPNUTY tak, že jsem příslušným hodnotám MHZ 0 a MLZ přiřadil 1 nebo 0, přičemž při zapnutí MH se v programu přiřadí všem MH 1 až MH 6 hodnota 1.

Tam, kde se v uživatelském programu předpokládá po ukončení výpočtu nové najetí programu po uplynutí určitého intervalu jsem provedl přerušení a zadání nových hodnot z PS. To umožňuje ponechání všech doposud nastavených a vypočtených hodnot a navíc zadání nových podmínek. Program v příl. č. 5.

4.4 Popis simulačních zkoušek

Simulační zkoušky byly prováděny tak aby vyzkoušely všechny nejčastěji předpokládanější stavy a aby se na tiskárnu vytiskly

všechna hlášení, které program obsahuje.

Při prvním průchodu byly zadány data jako při najetí ML ze studeného stavu. Při dotazu na PS "LIMIT NAJETÍ" jsem zadal, že MH nenajely, přesto, že proběhl limit najetí. Stav se nahlásil na tiskárně a výpočet se přerušil.

Druhý průchod jsem zadal tak, jako by žádná ML nebyla volná. Nahlásilo se "NELZE ZVÝŠIT VÝKON" a výpočet se přerušil. Při třetím průchodu jsem nastavil najeté MH. Provedl se tisk "START ML AUTOMATICKY". Výpočet se přerušil z dotazem "LIMIT OBSLUHY". Zadal jsem, že je limit splněn a vytiskne se "AUTOMATICKY". Výpočet se přerušil s dotazem "LIMIT ML". Zadal jsem, že limit vybrané ML je vyčerpán, k najetí však přece nedošlo. Tento stav byl hlášen tiskem. "LIMIT PŘEKROČEN U ML 1".

Čtvrtý průchod pracoval s hodnotami průchodu třetího. Výpočet se přerušil s dotazem "LIMIT ML". Zadal jsem, že k zapnutí ML došlo v limitu a na tiskárně se vytisklo "ML 1 ZAPNUTA". Výpočet pokračoval a přerušil se s dotazem "LIMIT PO NAJETÍ, ZVÝŠIT VÝKON". Zadal jsem, že limit je splněn a požadavek výkonu trvá. Na tiskárně se vytisklo "START ML AUTOMATICKY" a výpočet se přerušil s dotazem na PS "LIMIT ML" zadal jsem že najetí nebylo splněno v limitu. Na tiskárně došlo k tisku "ML 6 ZAPNUTA".

Dále jsem v předepsaném pořadí zapnul postupně všechny ML jejich zapnutí jsem signalizoval na tiskárně "tiskem" ML N zapnuta. Po vyčerpání všech linek došlo k hlášení "NELZE ZVÝŠIT VÝKON". Výpočet se přerušil.

V dalším průchodu jsem nastavil takové hodnoty, které simulovaly najetí havarii a vstoupily do programu přímo na automaticky provoz ML. Zadal jsem, že u všech linek je teplota za

třidičem větší jak 150° . Na tiskárně se napsalo "NAJETÍ PO HA-VARII" a výpočet se přerušil.

V posledním průchodu jsem ponechal nastavené hodnoty z předchozího, ale zadal jsem, že u ML 1 a ML 2 nedošlo k najetí mazutů a u ML 5 nejsou splněny podmínky najetí ML. Poté jsem postupně zapínal všechny linky po sobě. Linky u nichž bylo najetí možné se najely. Protože hodnoty byly zadány tak, aby došlo k nepřípustné konfiguraci toto hlášení se na-konec na tiskárně vytisklo.

Simulační zkoušky jsou uvedeny v příloze 6.

5 ALGORUTMUS K3-4 V JAZYKU KORAL

Systém koral byl sestaven pro přepis z vývojových diagramů. Pro přepis z RT by bylo třeba provést některé drobné úpravy. Protože tento úkol byl mimo rozsah mé práce, algoritmus K3-4 byl sestaven z vývojového diagramu.

5.1 Seznam použitých příkazů.

KOZA	konec zadání
KOPR	konec řídicího programu
KONE	výsledná dvouhodnotová proměnná je konjunkcí vstupů z možností negace vstupů
DINE	disjunkce s negací zvolených vstupů
DIKN	disjunkce konjunkcí s negací zvolených vstupů
SKOK	nepodmíněný
SCIS	skok číselně podmíněný
SLOG	skok logicky podmíněný
SKOP	opakování skoku
SRIZ	skok řízený hodnotou číselné proměnné definované příkazem UCIB
PREC	přepínač skoku

SOCE součet integrovaných proměnných
 SCIT součet realových proměnných
 SEVE sestavení vektoru
 UCIB nastavení cíle skoku pro příkaz SRIZ

5. 2 Program v jazyku koral

```
HLAV 13 K3-4 1 10 13 1 XX4
;   NAJIZDENI A PROVOZ MH A ML
MODE
MODE
PREC 255X 1X 151X RPK3 0
151X PREC 255X 201X 255X EPK3 1
1X KONE ZNK1 = -SKL1
16X KONE ZNK 2 = -QZH3
20X DIKN L42=/SKL1xQZH3xNHK3x-AOP1x-AOP2x-AOP3
                                     -AOP4x-AOP5x-AOP6/
      /QZL4xPA1x-AOP1x-AOP2x-AOP3
      x-AOP4x-AOP5x-AOP6/
      /QZL4xPA2x-AOP1x-AOP2x-AOP3
      x-AOP4x-AOP5x-AOP6/
      SLOG 255X 141X L42
20H   KONE HL2=XX0
246X  UCIB C9 = 151X
      SOCE RPK3 = 0
KONE   XX4=XX0
255X   KOPR
21H    SOCE   RPK3=1
      KOPR
201X   SRIZ   C9
141X   KONE   HL1=XX0
      KONE   HL4=XX0
```

KONE HL5=XXO
KONE HL6=XXO
KONE HL7=XXO
KONE HL8=XXO
KONE HL9=XXO
KONE HL10=XXO
KONE HL11=XXO
KONE HL12=XXO
KONE HL13=XXO
KONE HL14=XXO
KONE HL15=XXO
KONE HL16=XXO

SLOG 245X 20H HL17

245X SCIT C11 =0

SCIT C12=0

SCIT C13=0

SCIT C14=0

SCIT C15=0

SCIT C16=0

SOCE P7=0

SOCE P8=0

SLOG 30X 31X MZK1

31X SOCE P7=1+P7

SKOK 32X

30X SCIT C11=1

32X SLOG 33X 34X MZK6

34X SOCE P8=1+P8

SKOK 35X

33X SCIT C12=1

35X SLOG 36X 37X MZK3

37X SOCE P7=1+P7
 SKOK 38X
 36X SCIT C13=1
 38X SLOG 39X 40X MZK4
 40X SOCE P8=1+P8
 SKOK 41X
 39X SCIT C14=1
 41X SLOG 42X 43X MZK2
 43X SOCE P7=1+P7
 SKOK 44X
 42X SCIT C15=1
 44X SLOG 45X 46X MZK5
 45X SCIT C16=1
 SKOK 240X
 46X SOCE P8=1+P8
 240X SOCE C32=P7+P8
 PREC 7X 8X 8X C32 3
 7X SLOG 241X 12X MHZO
 241X SLOG 242X 255X PA3
 242X KONE NAH1=XX1
 11X SKOP A1=9X 10X 24 0
 9X UCIB C9=11X
 SKOK 21H
 10X SLOG 243X 12X MHZO
 243X KONE HL16=XX1
 KONE PA3=XX1
 SKOK 20H
 12X SOCE A1=0
 SOCE RPK3 = 0
 KONE PA2=XX1

SKOK 13X
 8X KONE PA1=XX1
 13X SLOG 255X 5X QZL4
 5X KONE L10=MZK1xMZK2xMZK3xMZK4xMZK5xMZK6
 SLOG 51X 52X L10
 52X SLOG 53X 54X QZL4
 54X KONE HL1=XX1
 53X SKOK 20H
 51X KONE HL2=XX1
 57X SKOP A18=47X 56X 6 0
 47X UCIB C9=57X
 SKOK 21H
 56X SLOG 48X 246X OV1
 48X SOCE C30=P7-P8
 PREC 21X 22X 27X C30 0
 22X SOCE C6=0
 SEVE P6=C12 1 C11 1 C14 1 C15 1 C16 1 C13 1
 23X SOCE C6=1+C6
 PREC 34X 24X 52X C6 6
 24X VYPR C7=P6 C6
 SCIS 23X 26X C7 0.5
 26X PREP C6 73X 72X 75X 76X 77X 74X
 72X SCIT C11=0
 SLOG 160X 161X MHZ1
 160X PREC 78X 161X 161X C32 2
 161X CLOF L21=AML1 xQZL4xSVPlx-/MLZ1+AOPl/
 SLOG 78X 79X L21
 78X KONE HL4=XX1
 SKOK 180X
 79X KONE NALL=XX1

82X SLOG 80X 135X MLZ1
 80X SKOP A19=83X 84X 30 0
 83X UCIB C9=82X
 SKOK 21H
 84X KONE HL10=XX1
 SKOK 180X
 73X SCIT C12=0
 SLOG 162X 163X MHZ6
 162X PREC 88X 163X 163X C32 2
 163X OLOF L26=AML6xQZL4xSVP6x-/MLZ6+AOP6/
 SLOG 88X 89X L26
 88X KONE HL5=XX1
 SKOK 180X
 89X KONE NAL6=XX1
 92X SLOG 90X 136X MLZ6
 90X SKOP A20=93x 94X 30 0
 93X UCIB C9=92X
 SKOK 21H
 94X KONE HL11=XX1
 SKOK 180X
 74X SCIT C13=0
 SLOG 164X 165X MHZ3
 164X PREC 98X 165X 165X C32 2
 165X OLOF L23=AML3xQZL4xSVP3x-/MLZ3+AOP3/
 SLOG 98X 99X L23
 98X KONE HL6=XX1
 SKOK 180X
 99X KONE NAL3=XX1
 102X SLOG 100X 135X MLZ3
 100X SKOP A21=103X 104X 30 0

103X UCIB C9=102X
 SKOK 21H
 104X KONE HL12=XX1
 SKOK 180X
 75X SCIT C14=0
 SLOG 166X 167X MHZ4
 166X PREC 108X 167X L67X C32 2
 167X OLOF L24=AML4xQZL4xSVP4x-/MLZ4+AOP4/
 SLOG 108X 109X L24
 108X KONE HL7=XX1
 SKOK 180X
 109X KONE NAL4=XX1
 112X SLOG 110X 136X MLZ4
 110X SKOP A22=113X 114X 30 0
 113X UCIB C9=112X
 SKOK 21H
 114X KONE HL13=XX1
 SKOK 180X
 76X SCIT C15=0
 SLOG 168X 169X MHZ2
 168X PREC 118X 169X 169X C32 2
 169X OLOF L22=AML2xQZL4xSVP2x-/MLZ2+AOP2/
 SLOG 118X 119X L22
 118X KONE HL8=XX1
 SKOK 180X
 119X KONE NAL2=XX1
 122X SLOG 120X 135X MLZ2
 120X SKOP A23=123X 124X 30 0
 123X UCIB C9=122X
 SKOK 21H

124X KONE HL14=XX1
 SKOK 180X
 77X SCIT C16=0
 SLOG 170X 171X MHZ5
 170X PREC 128X 171X 171X C32 2
 171X OLOF L25=AML5xQZL4xSVP5x-/MLZ5+AOP5/
 SLOG 128X 129X L25
 128X KONE HL9=XX1
 SKOK 180X
 129X KONE NAL5=XX1
 132X SLOG 130X 136X MLZ5
 130X SKOP A24=133X 134X 30 0
 133X UCIB C9=132X
 SKOK 21H
 134X KONE HL15=XX1
 SKOK 180X
 135X SOCE P7=1+P7
 SOCE A19 = 0
 SOCE A21 = 0
 SOCE A 23=0
 SKOK 150X
 136X SOCE P8=1+P8
 SOCE A20=0
 SOCE A22=0
 SOCE A24=0
 150X SLOG 20H 138X QZL4
 138X SCIT C17=C11+C12+C13+C14+C15+C16
 SCIS 20H 182X C17 0.5
 182X SKOP A2=183X 51X 18 0
 183X UCIB C9=182X

SKOK 21H
 180X SLOG 20H 181X QZL4
 181X SCIT C17=C11+C12+C13+C14+C15+C16
 SCIS 20H 51X C17 0.5
 27X SOCE C6=0
 SEVE P30=C14 1 C16 1 C12 1
 210X SOCE C6=1+C6
 PREC 211X 211X 212X C6 3
 211X VYPR C7=P30 C6
 SCIS 210X 214X C7 0.5
 214X PREP C6 75X 77X 73X
 212X PREC 216X 234X 216X P8 2
 234X PREC 216X 218X 216X P7 1
 216X PREC 219X 215X 219X P8 1
 215X PREC 219X 218X 219X P7 0
 219X PREC 220X 235X 220X P8 0
 235X PREC 220X 218X 220X P7 2
 220X PREC 21X 218X 21X P7 3
 21X SOCE C6=0
 SEVE P31=C11 1 C15 1 C13 1
 225X SOCE C6=1+C6
 PREC 226X 226X 227X C6 3
 226X VYPR C7=P31 C6
 SCIS 225X 224X C7 0.5
 224X PREP C6 72X 76X 74X
 227X PREC 229X 236X 229X P8 1
 236X PREC 229X 218X 229X P7 2
 229X PREC 221X 237X 221X P8 0
 237X PREC 221X 218X 221X P7 1
 221X PREC 230X 232X 230X P7 0

232X PREC 230X 218X 230X P8 2
230X PREC 27X 218X 27X P8 3
218X KONE HL17=XX1

SKOK 20H

KOZA

6 Zhodnocení návrhu řízení a porovnání s obvody pevné logiky.

Pomocí simulačních zkoušek jsem ověřil, že návrh řízení je po logické stránce správný.

Číslicové řízení výroby je nejprogresivnější způsob řízení. Oproti obvodům pevné logiky má velké množství výhod a to zvláště při projektování výrobních systémů, ale samozřejmě i při vlastním výrobním procesu.

Délka života výrobního systému se skládá ze tří částí:

- úvodní projekt
- konečné projektové řízení
- vlastní výroba

První dvě nevýrobní fáze trvají zhruba tři roky. U použití obvodů pevné logiky je třeba již tři roky před vlastní výrobou vědět o všech "drátech" jež budou použity, což samozřejmě prodlužuje nevýrobní fázi. Během tří let může dojít k některým změnám, a pak je třeba předělávat návrh logického řízení a u obvodů s pevnou logikou tak dochází ke značným časovým ztrátám. Navíc za tři roky dojde v dnešní době určitě k dalšímu vývoji a jeho poznatky již nelze v případě použití obvodů pevné logiky aplikovat.

Oproti tomu u číslicového řízení lze navrhnout na počátku projektu pouze hierarchii a hardware, ovšem vlastní způsob řízení software lze zpracovávat během projektového řešení, nebo až po něm. Tím samozřejmě odpadají popsané nevýhody obvodů pev-

né logiky. Zde je tedy technický i ekonomický efekt zřejmý.

Druhou výhodou je možnost opravnosti a to velmi jednoduše pouze perem. U obvodů pevné logiky se musí znovu přepájet některé "fousy". I vyhledávání místa, které se chce opravit je dosti náročné.

Použití číslicového řízení je tedy rychlejší, jednodušší a levnější. Tyto přednosti zvláště u neopakovatelných projektů.

Třetí výhoda použití číslicového řízení je v možnosti simulačních zkoušek. Tím dochází k šetření výrobního zařízení, značnému zkrácení doby zavedení systému řízení do výroby.

Další výhodou je univerzálnost počítače. Díky této vlastnosti lze použít počítač i při dalších činnostech souvisejících s výrobou. Jde zvláště o různé ekonomické činnosti. Navíc je zde jednodušší údržba a součástková základna je menší.

Výhody číslicového řízení vyniknou i při inovaci. U obvodů pevné logiky je třeba znovu projektovat, avšak u číslicového řízení se při inovaci vymění starý počítač za nový, ale nástroj řízení - software - zůstává, pouze se provedou "perem" změny /přepis do jiného programovacího jazyka/.

Je tedy zřejmé, že číslicové řízení je podstatně efektivnější, než použití obvodů pevné logiky. Při jeho aplikaci lze také v mnohem větším měřítku využívat spolupráce a to jak mezi jednotlivými podniky, tak i v měřítku mezinárodním. /Výměna softwarové dokumentace/. Při dalším vývoji, a tedy používáním modernějších počítačů a mikroprocesorů, bude docházet k dalšímu zmenšování rozměrů techniky a k větší spolehlivosti /možnost větší redundance/. Proto je cesta číslicového řízení jedinec správná a tedy nutná.

7 Závěr

Úloha systému logického řízení skupiny spalování je dílčí úlohou systému logického řízení celého elektrárenského bloku. Proto je zpráva psána tak, že je do tohoto systému začleněna. Uvedl jsem pojednání o současném stavu řízení elektrárenských bloků a o jeho perspektivách.

Při zpracování programu se ukázalo, že je velmi výhodné použití rozhodovacích tabulek. Nejen že se jejich použitím program zkrátil a jednodušeji odladil, ale hlavně práce s tabulkami je jednoduchá a přehledná.

Je proveden popis rozhodovacích tabulek a popis sestavení programu s jejich pomocí. Pro další využití práce jsem uvedl i nezbytnou teorii rozhodovacích tabulek, protože u nás tato technika není příliš známá a používaná. Dále jsem též uvedl výhody používání rozhodovacích tabulek, zvláště pak ve srovnání s vývojovými diagramy.

Zaměřil jsem se také na srovnání číslicového řízení a řízení s použitím obvodů pevné logiky. Jednoznačné výsledky tohoto srovnání jsou ve zprávě napsány.

Vlastní návrh logického řízení jsem kontroloval simulačními zkouškami, jež jsou ve zprávě popsány a uvedeny.

Této práce bude prakticky využito při inovaci řídicího systému elektrárenských bloků 110 a 200 MW v druhé polovině osmdesátých let.

LITERATURA

1. B. Pidermann

Zkoušky číslíkových algoritmů logického řízení na ověřovacím pracovišti ETU II

Zpráva EGU č. 12 20 1 10, 1979

2. B. Pidermann

Automatizace systému řízení elektrárenských bloků velkých výkonů v ČSSR

Zpráva EGU, 1979

3. Kolektiv

Uživatelská příručka programovacího systému pro realizaci algoritmů číslíkového řízení KORAL

Příloha zprávy EGU č. 12 30 1 05, 1978

4. J. Kešner

RT a jejich využití v budování ASŘ

Výběr informací z organizační a výpočetní techniky č.3,4, 5,6,1976

5. V. Chvalovský

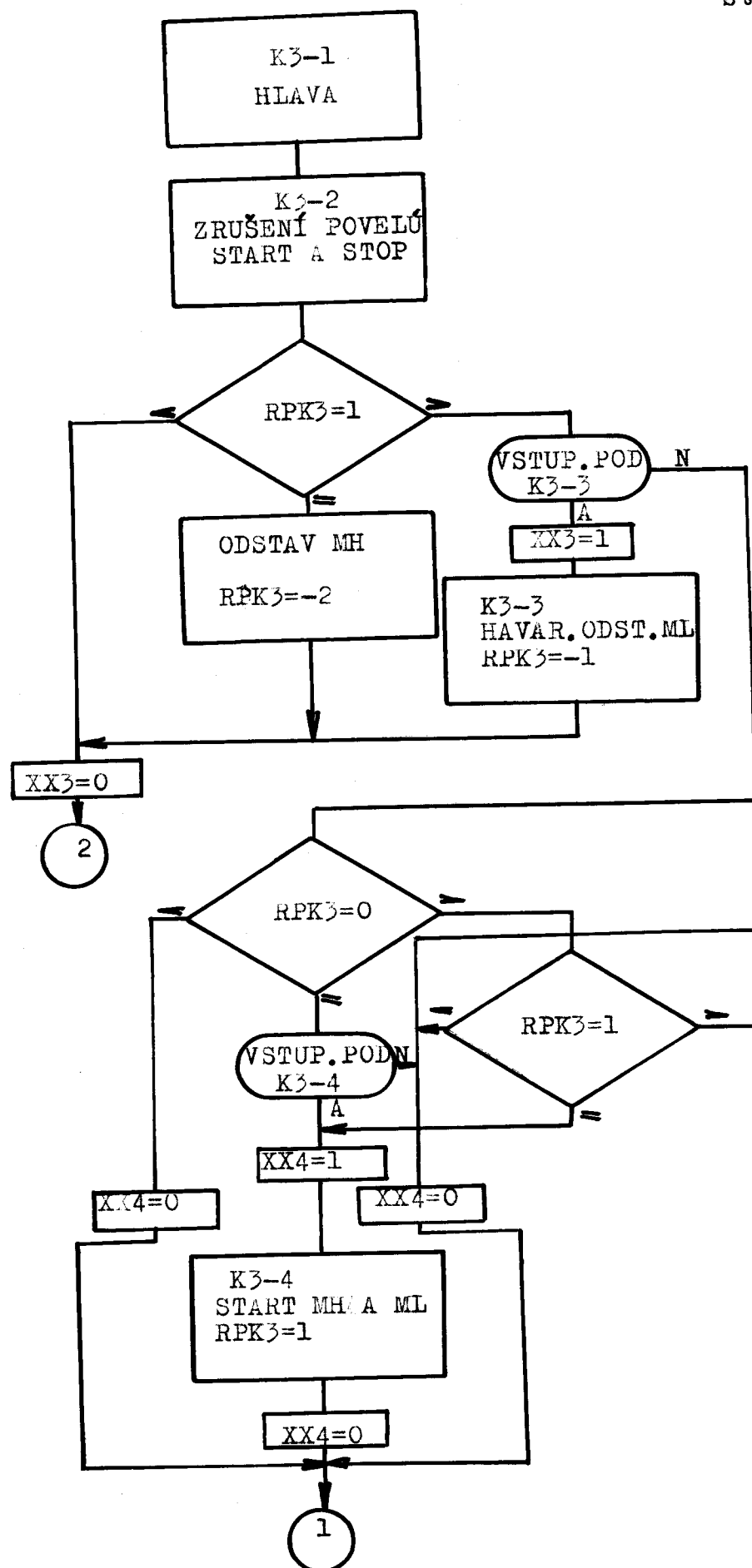
Rozhodovací tabulky

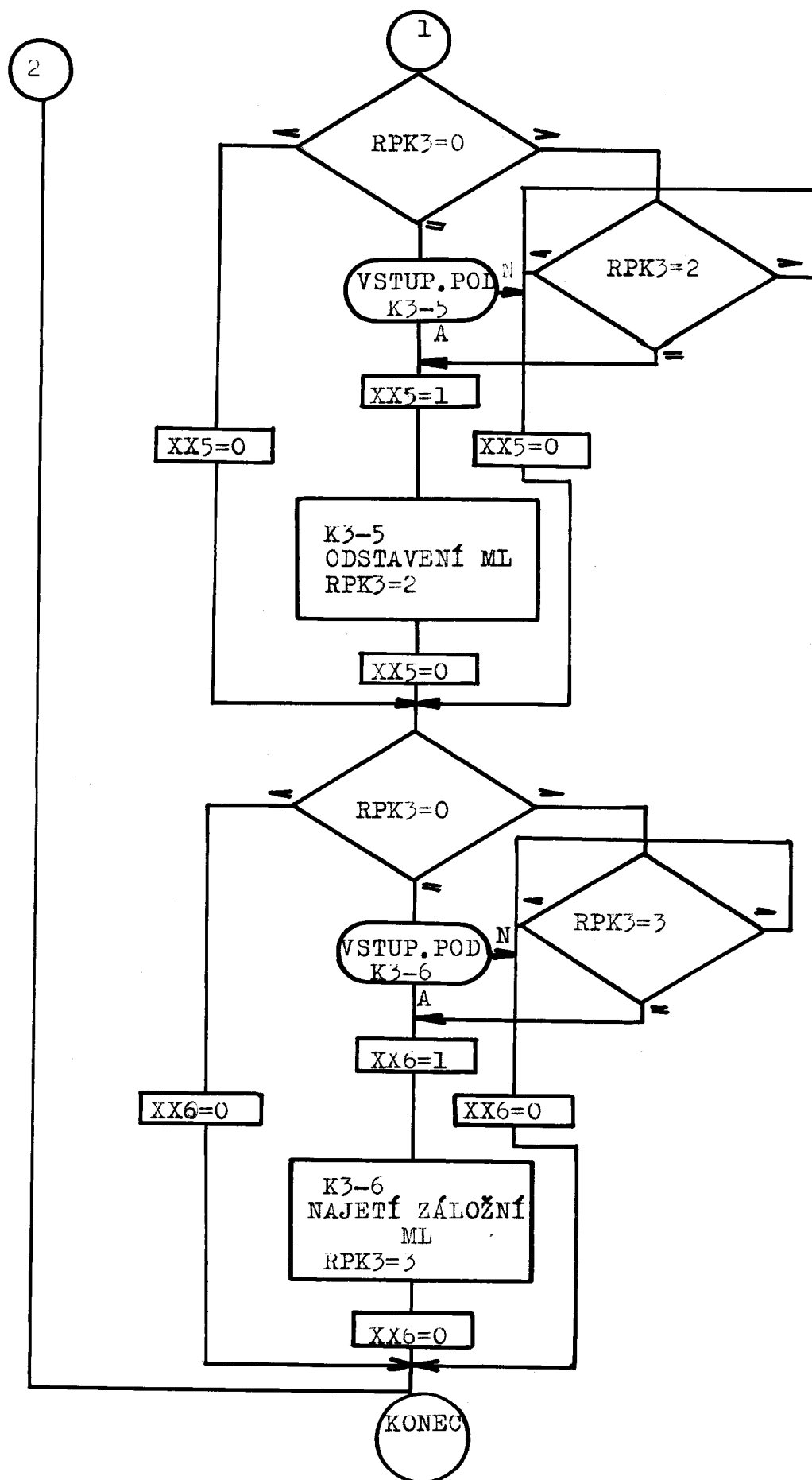
SNTL Praha, 1974

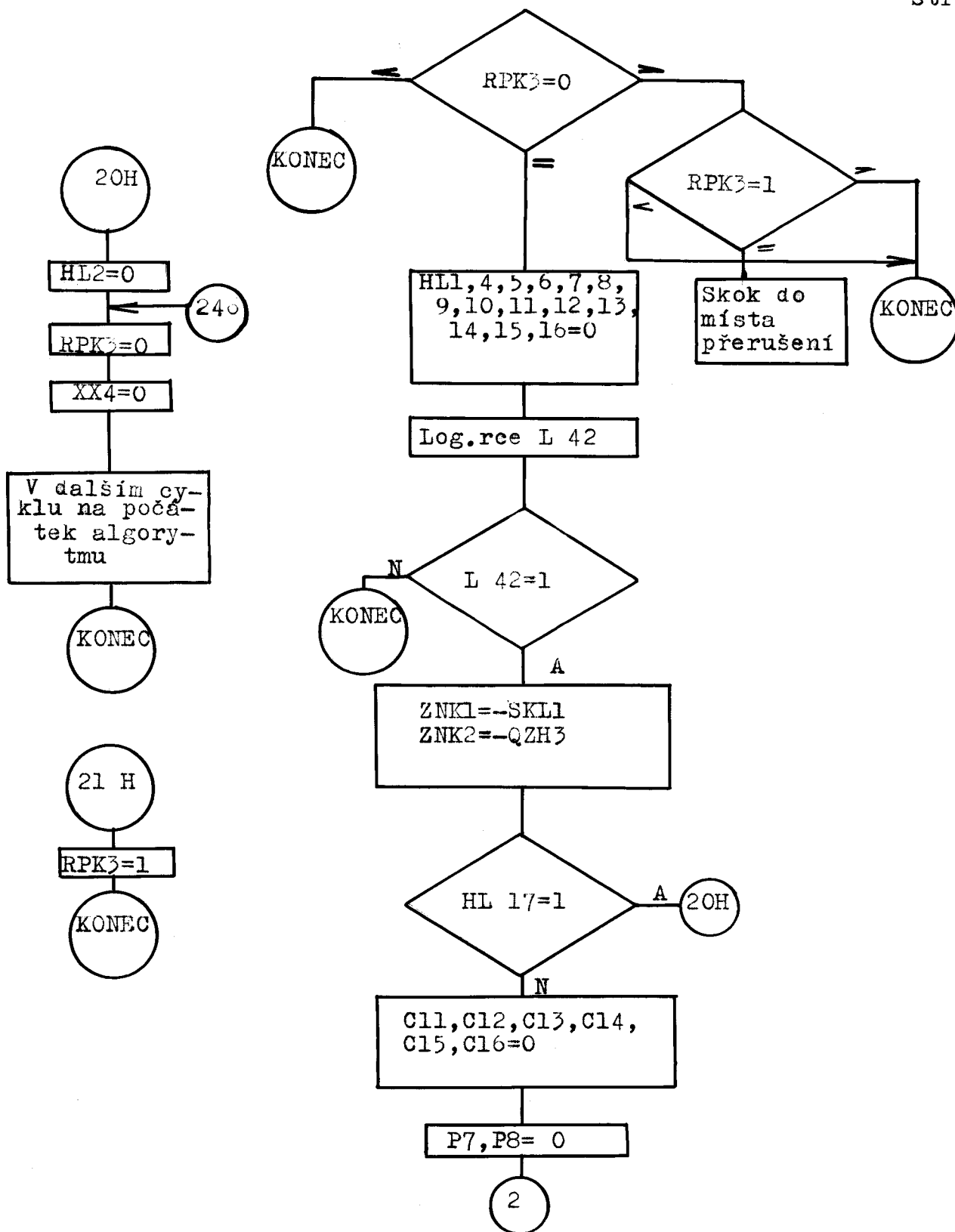
6. Kolektiv

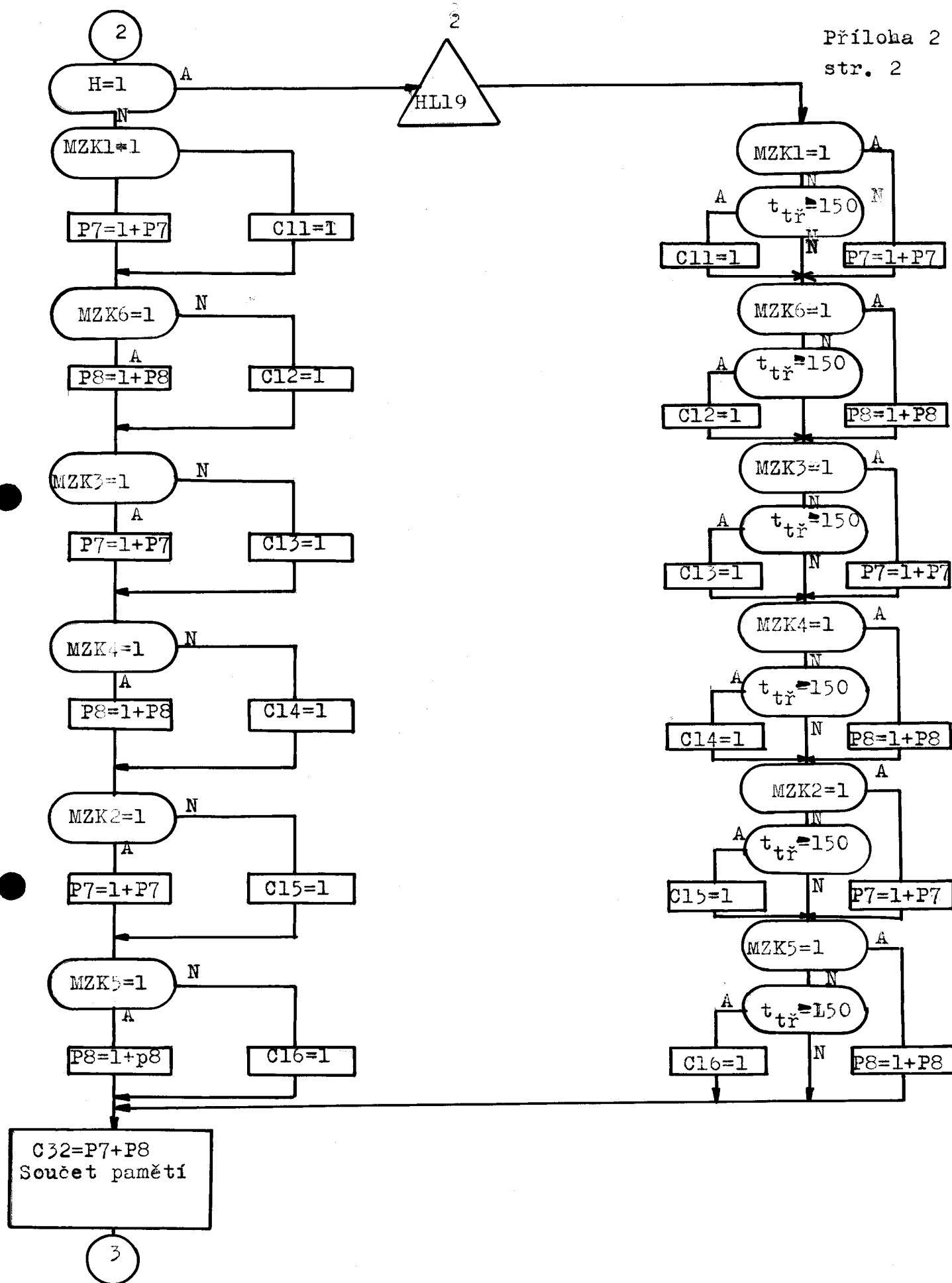
Programovací jazyk RPP - FORTRAN

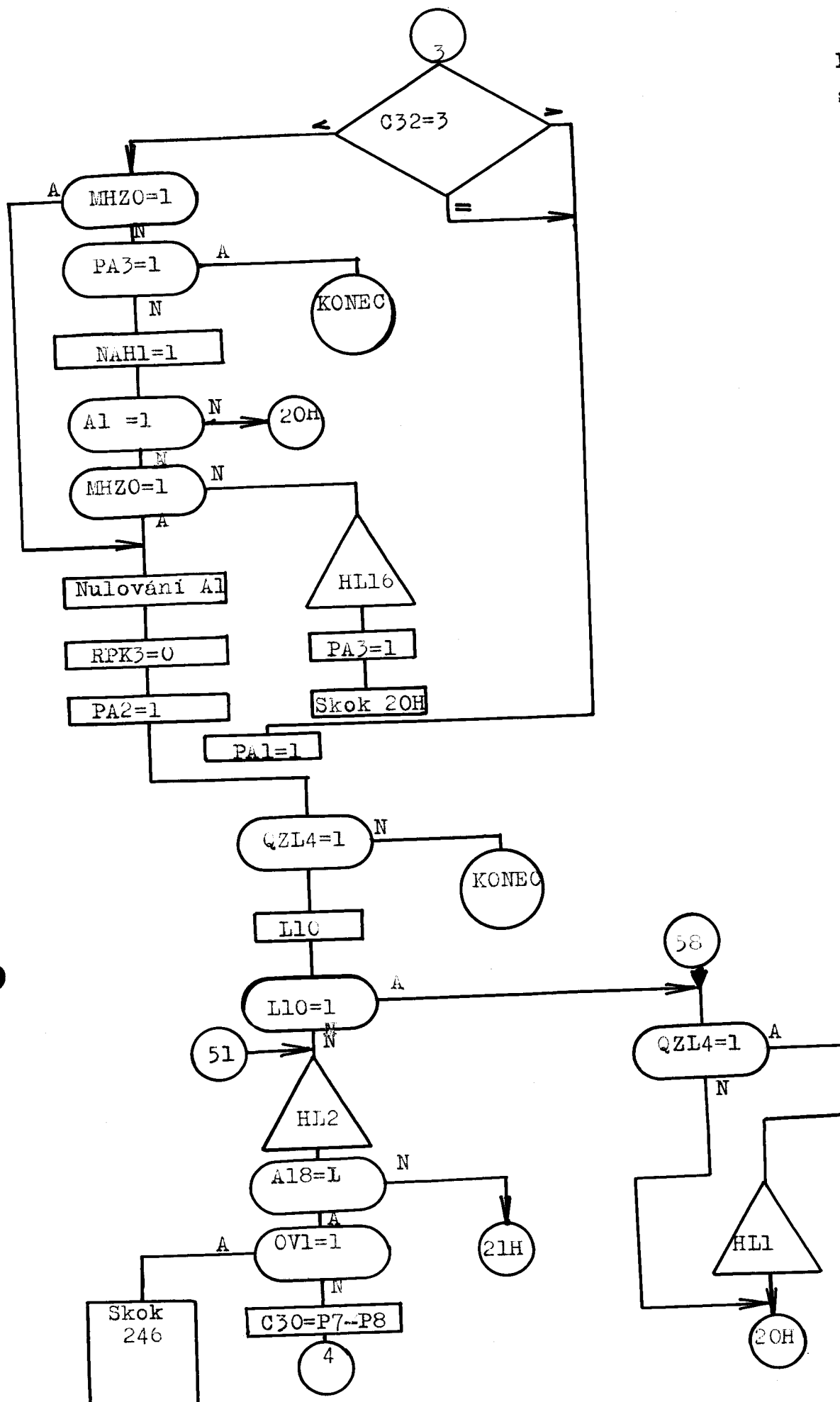
SNTL Praha, 1975

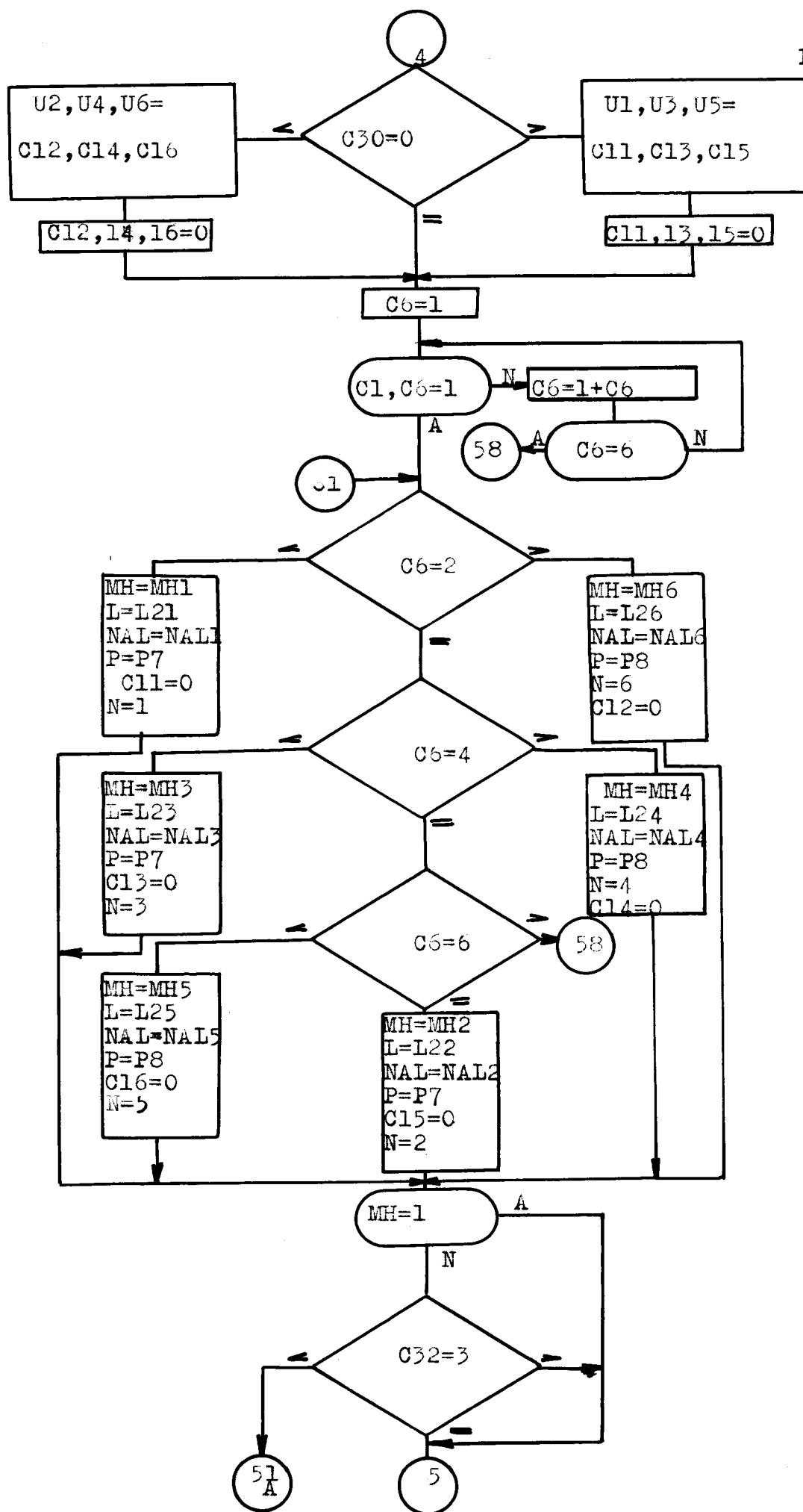


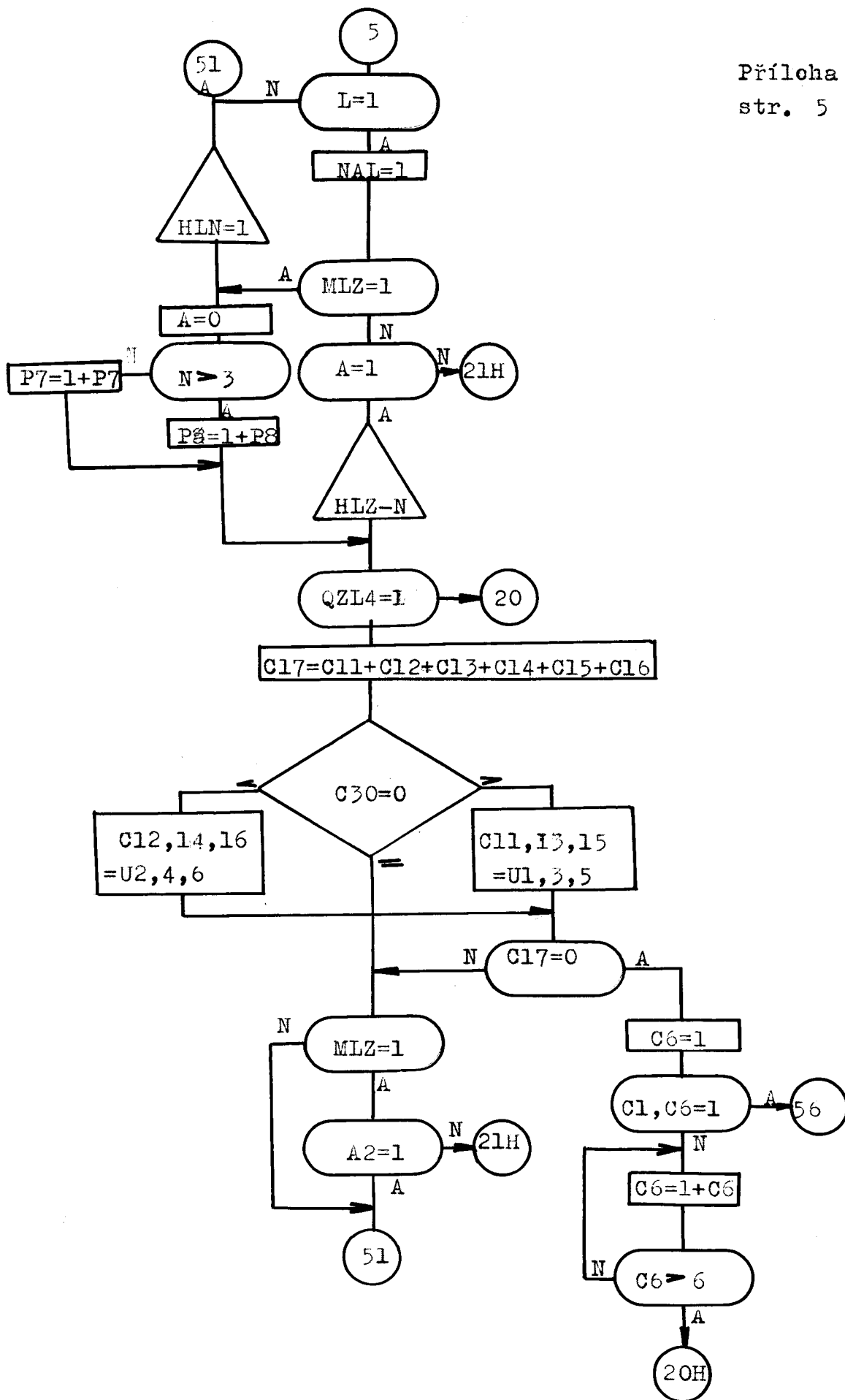


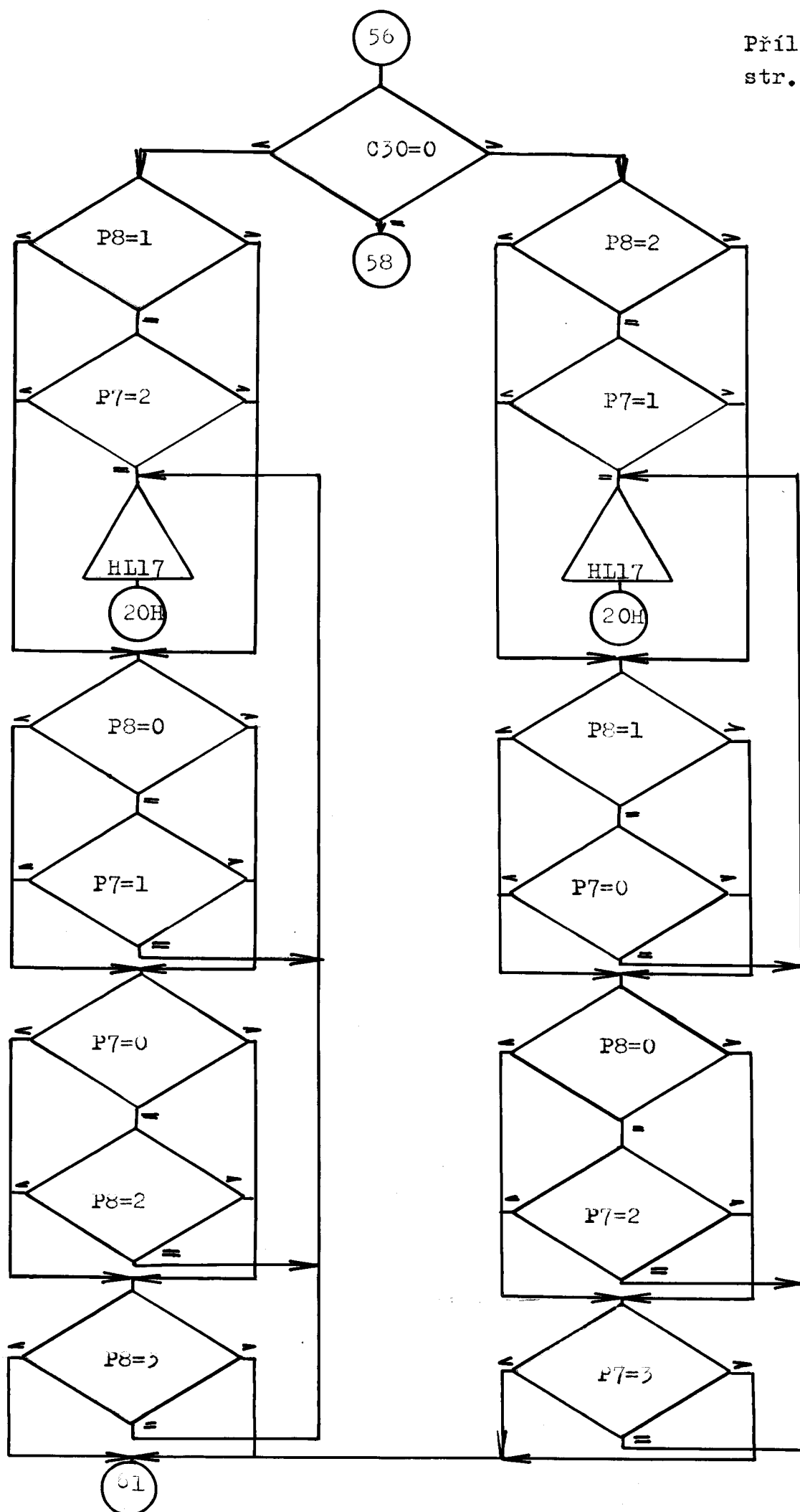


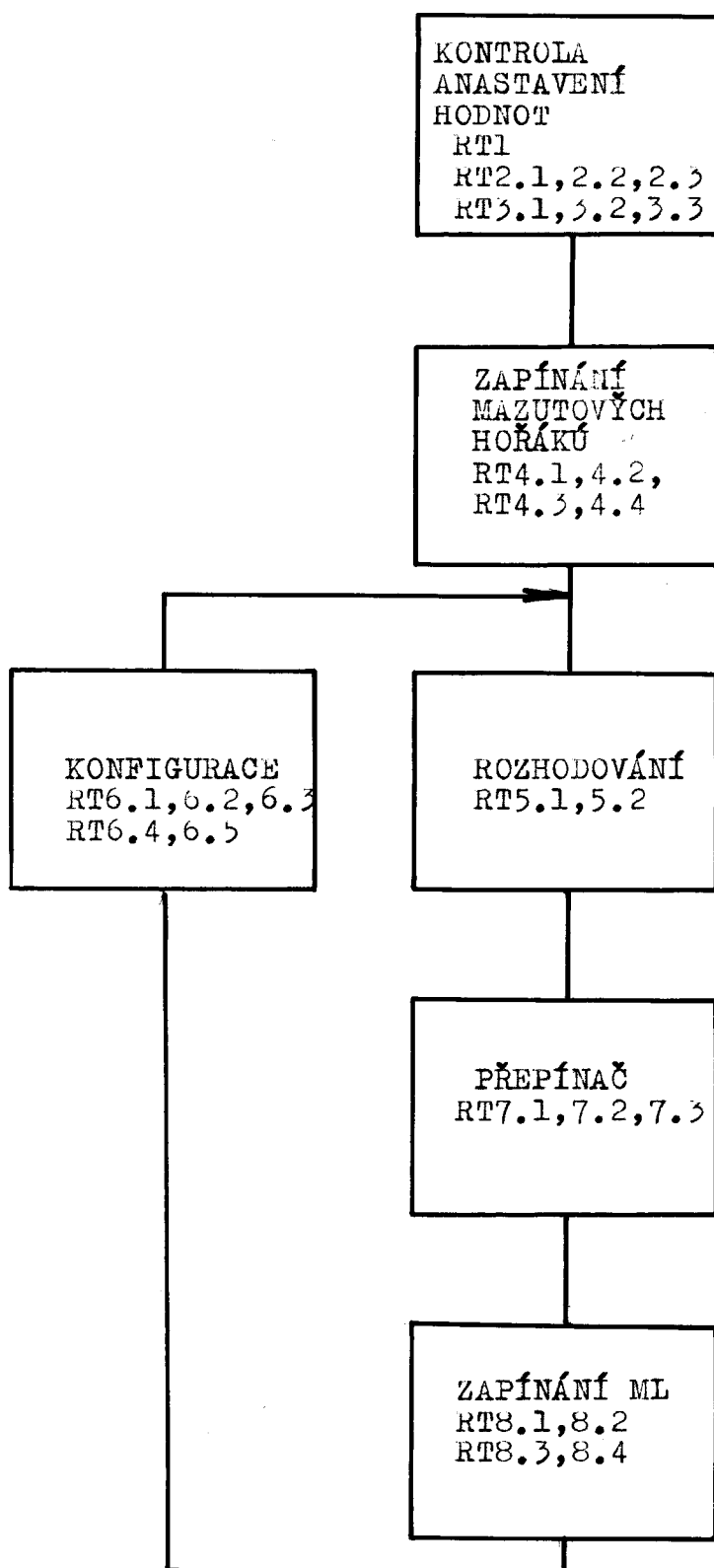












RT 1						
RPK 3 = 0	N	N	A	A	A	A
RPK 3 = 1	N	A	N	N	N	N
L 42 = 1	-	-	N	A	A	A
HL 17 = 1	-	-	-	A	N	N
H = 1	-	-	-	-	N	A
KONEC	X		X	X		
ZNK 1 = -SKL 1			X	X	X	X
ZNK 2 = -QZH 3			X	X	X	X
HL 1,4-16 = 0				X	X	X
C 11 - 16 = 0					X	X
P 7, P 8 = 0					X	X
HL 19 = 1						X
GOTO RT 2. 1					X	
GOTO RT 3. 1						X
GOTO RT 5. 1		X				
GOTO RT 20 H				X		

RT 2. 1				
MZK 1 = 1	A	A	N	N
MZK 6 = 1	A	N	N	A
P 7 = 1 + P 7	X	X		
P 8 = 1 + P 8	X			X
C 11 = 1			X	X
C 12 = 1		X	X	
GOTO RT 2. 2	X	X	X	X

RT 2. 2				
MZK 3 = 1	A	A	N	N
MZK 4 = 1	A	N	N	A
P 7 = 1 + P 7	X	X		
P 8 = 1 + P 8	X			X
C 13 = 1			X	X
C 14 = 1		X	X	
GOTO RT 2. 3	X	X	X	X

RT 2. 3				
MZK 2 = 1	A	A	N	N
MZK 5 = 1	A	N	N	A
P 7 = 1 + P 7	X	X		
P 8 = 1 + P 8	X			X
C 15 = 1			X	X
C 16 = 1		X	X	
C 32 = P7 + P8	X	X	X	X
GOTO RT 5. 1	X	X	X	X

RT 20 H	
HL 2 = 0	X
RPK 3 = 0	X
XX 4 = 0	X
ČEKEJ 10 SEC	X
GOTO RT 1	X

RT 3. 1									
MZK 1 = 1	A	A	A	N	N	N	N	N	N
Ttř 1 $\cong$ 150	-	-	-	A	N	A	A	N	N
MZK 6 = 1	A	N	N	A	A	N	N	N	N
Ttř 6 $\cong$ 150	-	A	N	-	-	A	N	A	N
P 7 = 1 + P 7	X	X	X						
P 8 = 1 + P 8	X			X	X				
C 11 = 1				X		X	X		
C 12 = 1		X				X		X	
GOTO RT 3. 2		X	X	X	X	X	X	X	X

[illegible]

RT 3. 3									
MZK 2 = 1	A	A	A	N	N	N	N	N	N
Ttř 2 $\geq$ 150	-	-	-	A	N	A	A	N	N
MZK 5 = 1	A	N	N	A	A	N	N	N	N
Ttř 5 $\geq$ 150	-	A	N	-	-	A	N	A	N
P 7 = 1 + P 7	X	X	X						
P 8 = 1 + P 8	X			X	X				
C 15 = 1				X		X	X		
C 16 = 1		X				X		X	
C 32 = P7+P8	X	X	X	X	X	X	X	X	X
GOTO RT 4. 1	X	X	X	X	X	X	X	X	X

RT 4. 1			
C 32 $\geq$ 3	N	N	N
MHZ 0 = 1	N	N	A
PA 3 = 1	A	N	-
KONEC	X		
NAH 1 = 1		X	
PA 1 = 1			X
A 1 = 0			X
RPK 3 = 0			X
PA 2 = 1			X
GOTO RT 4. 3		X	X
GOTO RT 4. 2		X	

RT 4. 2			
A 1 = 1	N	A	A
MHZ 0 = 1	-	A	N
A 1 = 0		X	
PA 2 = 1		X	
H1 16 = 1			X
PA 3 = 1			X
RPK 3 = 0		X	
GOTO RT C 9	X		
GOTO RT 4. 3		X	
GOTO RT 20 H			X

RT 4. 3			
QZL 4 = 1	N	A	A
L 10 = 1	-	N	A
KONEC	X		
HL 2 = 1		X	
HL 1 = 1			X
GOTO RT 20 H			X
GOTO RT 4. 4		X	

RT C 9	
RPK 3 = 1	X
ČEKEJ 10 SEC	X
NÁVRAT	X

RT 4. 4			
A 18 = 1	N	A	A
OV 1 = 1	-	A	N
KONEC		X	
RPK 3 = 0		X	
XX 4 = 0		X	
C 30 = P7 - P8			X
GOTO RT 5. 1			X
GOTO RT C 9	X		

RT 5. 1			
C 30 = 0	<	=	>
U2,4,6 = C 12,14,16	X		
C 12, 14,16 = 0	X		
U 1, 3, 5, = C 11,13,15			X
C 11,13,15 = 0			X
C 6 = 1	X	X	X
GOTO RT 5.2	X	X	X

RT 5. 2			
C 1, C 6 = 1	A	N	N
C 6 > 6	-	N	A
C 30 = 0			X
C 6 = 1 + C 6		X	
GOTO RT 5. 2		X	
GOTO RT 7. 1	X		
GOTO RT 6. 1			X

RT 6. 1				
C 30 : 0	≡	=	<	>
QZL 4 = 1	A	N		
HL 1 = 1	X			
GOTO RT 20 H	X	X		
GOTO RT 6. 4			X	
GOTO RT 6. 2				X

RT 6. 2					
P 8 = 2	A	N	N	A	N
P 7 = 1	A	-	-	N	-
P 8 = 1	-	A	N	N	A
P 7 = 0	-	A	-	-	N
HL 17 = 1	X	X			
GOTO RT 20 H	X	X			
GOTO RT 6. 3			X	X	X

RT 6. 3					
P 8 = 0	A	A	A	N	N
P 7 = 2	A	N	N	-	-
P 7 = 3	-	A	N	A	N
HL 17 = 1	X	X		X	
GOTO RT 20 H	X	X		X	
GOTO RT 7. 1			X		X

RT 6. 4					
P 8 = 1	A	N	N	A	N
P 7 = 2	A	-	-	N	-
P 8 = 0	-	A	N	N	A
P 7 = 1	-	A	-	-	N
HL 17 = 1	X	X			
GOTO RT 20 H	X	X			
GOTO RT 6. 5			X	X	X

RT 6. 5					
P 7 = 0	A	A	A	N	N
P 8 = 2	A	N	N	-	-
P 8 = 3	-	A	N	A	N
HL 17 = 1	X	X		X	
GOTO RT 20 H	X	X		X	
GOTO RT 7. 1			X		X

RT 7. 1			
C6 : 2	<	=	>
MH = MH 1	X		
L = L 21	X		
NAL = NAL 1	X		
P = P 7	X		
C 11 = 0	X		
N = 1	X		
C 12 = 0		X	
MH = MH 6		X	
L = L26		X	
NAL = NAL 6		X	
P = P 8		X	
N = 6		X	
GOTO RT 8. 1	X	X	
GOTO RT 7. 2			X

RT 7. 2			
C6 : 4	<	=	>
MH = MH 3	X		
L = L 23	X		
NAL = NAL 3	X		
P = P 7	X		
C 13 = 0	X		
N = 3	X		
C 14 = 0		X	
MH = MH 4		X	
L = L 24		X	
NAL = NAL 4		X	
P = P 8		X	
N = 4		X	
GOTO RT 8. 1	X	X	
GOTO RT 7. 3			X

RT 8. 2				
QZL 4 = 1	N	A	A	A
C 30 = 0	-		=	
C 17		X	X	X
C 12,14,16= U 2,4,6		X		
C 11, 13, 15 = U 1, 3, 5				X
GOTO RT 20 H	X			
GOTO RT 8. 3		X	X	X

RT 8. 3				
C 17 $\geq$ 1	N	N	N	A
MLZ = 1	N	A	A	-
A 2 = 1	-	A	N	-
HL 2 = 1	X	X		
C 6 = 1				X
GOTO RT C 9			X	
GOTO RT 4. 4	X	X		
GOTO 8. 4				X

RT 8. 4			
C 1 , C 6	A	N	N
C 6 > 6	-	A	N
C 6 = 1 + C6			X
GOTO RT 20 H		X	
GOTO RT 6. 1	X		
GOTO RT 8. 4			X

RT 7. 3			
C 6 = 6	<	=	>
MH = MH 5		X	
L = L 25		X	
NAL = NAL 5		X	
P = P 8		X	
C 16 = 0		X	
N = 5		X	
C 15 = 0	X		
MH = MH 2	X		
L = L 22	X		
NAL = NAL 2	X		
P = P 7	X		
N = 2	X		
GOTO RT 8. 1	X	X	X

RT 8. 1						
MH = 1	N	-	-	-	-	-
C 32 $\geq$ 3	N	A	A	A	A	A
L = 1	-	N	A	A	A	A
MLZ = 1	-	-	A	A	N	N
N > 3	-	-	N	A	-	-
A = 1	-	-	-	-	N	A
P 7 = 1 + P 7			X			
P 8 = 1 + P 8				X		
HL 4 = 1	X	X				
NAL = 1			X	X	X	X
A = 0						X
HL /N./ = 1					X	
GOTO RT C9	X	X	X	X		X
GOTO RT 8.2						