

Донбаська державна машинобудівна академія
Кафедра автоматизації виробничих процесів

Цифрові системи управління й обробки інформації

Розділ 1

Організація й програмування систем ЧПУ Конспект лекцій

(для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”)

Краматорськ 2018

Цифрові системи управління й обробки інформації. Конспект лекцій.
Розділ 1: Організація й програмування систем ЧПУ. (для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”) / Укл. О. О. Сердюк. - Краматорськ: ДДМА, 2018. - 126 с.

Викладаються особливості проектування сучасних систем ЧПУ. Висвітлюється архітектурна організація систем ЧПУ. Приводяться методики проектування інформаційної моделі та приводних систем ЧПУ й розробки програмного забезпечення систем ЧПУ. Викладаються особливості побудови приводних систем для верстатів із ЧПУ.

Укладач

О. О. Сердюк, доц.

ЗМІСТ

1 ОСОБЛИВОСТІ ПРОЕКТУВАННЯ СУЧАСНИХ СИСТЕМ ЧПУ	5
1.1 Завдання числового програмного управління верстатами	5
1.2 Варіанти архітектурної організації сучасних систем ЧПУ	8
1.3 Варіанти реалізації відкритої архітектури систем ЧПУ	15
1.4 Організація зв'язків між компонентами системи керування	19
1.5 Особливості реалізації стандартів у системах ЧПУ	21
1.6 Реалізація інтерфейсних OPC у системах ЧПУ	23
1.7 Сутність виробничих стандартів STEP	28
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ МОДЕЛІ	25
2.1 Керування процесами в операційній системі ЧПУ	25
2.2 Склад інформаційної моделі	30
2.3 Система ЧПУ й об'єкт управління як функціональний автомат	32
2.4 Синтаксичний аналіз управляючої програми системи ЧПУ	33
2.5 Аналіз кадру управляючої програми	36
2.6 Приклад проектування управляючої таблиці МП-автомата	37
3 МЕТОДИ АЛГОРИТМІЗАЦІЇ ЗАВДАНЬ УПРАВЛІННЯ АВТОМАТИКОЮ	40
3.1 Застосування методу маскування	40
3.2 Метод маскування багатомісних логічних функцій	42
3.3 Формалізм опису складних автоматичних циклів	43
3.4 Графічне представлення паралельних процесів мережею Петрі	48
3.5 Формальне визначення мережі Петрі	50
3.6 Застосування мереж Петрі для моделювання	52
3.7 Розробка мережі Петрі для моделювання циклу автоматичної зміни інструмента	53
3.8 Моделювання процесу керування гнучкими виробничими модулями (ГВМ)	55
4 МОДЕРНІЗАЦІЯ СИСТЕМ ЧПУ	60
4.1 Аналіз цілей і завдань модернізації	60
4.2 Модернізація верстатів ЧПУ на базі систем SINUMERIK	64
4.3 Розробка структурної схеми системи ЧПУ верстата і її конфігурування	73
4.4 Розробка алгоритмів програмного забезпечення	80
5 ЗАГАЛЬНА ХАРАКТЕРИСТИКА СТРУКТУРИ ТА КОМПОНЕНТІВ ПРИВІДНОЇ СИСТЕМИ SIMODRIVE	83
5.1 Загальна характеристика двигунів	83
5.2 Огляд датчиків	85
5.3 Огляд приводних модулів SIMODRIVE	89
5.4 Модулі живлення	95

6 ПРОЕКТУВАННЯ СТРУКТУРИ ПРИВОДА SIMODRIVE	98
6.1 Принципи вибору двигунів, датчиків і плат керування	98
6.2 Непряма реєстрація положення з аналоговим і цифровим інтерфейсами	100
6.3 Пряма реєстрація положення з аналоговим керуванням.....	102
6.4 Пряма реєстрація положення із цифровим керуванням	105
6.5 Вибір і підключення модулів структури привода.....	108
7 ОРГАНІЗАЦІЯ ПРИВODНОЇ СИСТЕМИ SINAMICS	111
7.1 Особливості приводів SINAMICS	111
7.2 Приводи SINAMICS для верстатів із ЧПУ	113
7.3 Інтерфейс DRIVE-CLIQ	114
7.4 Вимірювальні системи з інтерфейсом DRIVE-CLIQ	118
7.5 Двигуни для підключення до SINAMICS	120
7.6 Модулі живлення для приводів	122
7.7 Уведення й вивід сигналів автоматики	122
7.8 Порядок проектування приводної системи SINAMICS S120	125
ЛІТЕРАТУРА	126

1 ОСОБЛИВОСТІ ПРОЕКТУВАННЯ СУЧАСНИХ СИСТЕМ ЧПУ

1.1 Завдання числового програмного управління верстатами

Завдання програмного управління визначають специфікацію функцій, які повинна виконувати система управління.

У загальному випадку перелік цих функцій може представляти сотні позицій, однак більшість функцій формується як би автоматично, згідно зі сформованими уявленнями та досвіду проектування систем ЧПУ по порівняно невеликому переліку основних вимог.

Залежно від складності технологічних процесів і завдань управління системи ЧПУ можна розділити на чотири групи:

- прості системи з мінімальним набором функцій, які використовуються для управління автоматикою в маніпуляторах, пилках, підйомниках, штампувальних молотах і ін.;
- системи зі стандартним набором функцій, наприклад, пристрої ЧПУ для токарських, фрезерних, свердлильних, шліфувальних верстатів, у яких передбачені стандартні програмні й апаратні засоби, які задовольняють більшості вимог споживачів;
- системи із гнучкими можливостями для унікального встаткування, які дозволяють управляти десятками координат, створити оптимальну конфігурацію системи керування й відрізняються широким спектром функцій;
- системи, об'єднані в локальні обчислювальні мережі з декількома операційними системами, призначені для розв'язку завдань комплексної автоматизації ділянки або цеху.

Приступаючи до проектування системи програмного управління, необхідно, у першу чергу, вивчити технологічні процеси, проаналізувати їхні особливості й недоліки, які слід усунути, сформулювати вимоги до технічних і програмних засобів.

На основі підготовлених вимог формується структура завдань ЧПУ, що в загальному випадку зводиться до наступного.

1. **Термінальне завдання.** Воно містить у собі ряд функцій:
 - уведення управляючих програм, аналіз їх коректності, редагування й зберігання програм;
 - діалог з оператором, завдання необхідних режимів роботи;
 - тестування й діагностика;
 - забезпечення зв'язку й взаємодії з іншими системами.
2. **Логічне завдання.** Це завдання містить у собі функції:

- управління автоматикою (кількість пристроїв автоматики на встаткування середньої складності досягає декількох сотень одиниць);
- збір і обробка інформації від датчиків про стан об'єкта;
- читання пульта, вивід повідомлень на пульт.

3. **Геометричне завдання.** Це завдання включає функції програмного керування координатними рухами, регулювання параметрів технологічної системи, компенсація нелінійностей і ін.

Розв'язок кожного завдання має деякі особливості для систем ЧПУ різних варіантів архітектурної організації.

Мови програмного вирішення завдань

У системах ЧПУ використовуються три мови:

- мова завдань – це кнопки, перемикачі й регулятори пульта оператора;
- мова задач – це стандартна система кодування управляючої програми під назвою G-код, в радянські часи – код ISO-7bit.
- мова дисплея – повідомлення, які видаються операторові (тексти, графіки, цифри, символи й т.п.).

Мова завдань пульта оператора й мова дисплея (людино-машинний інтерфейс) повинні забезпечити запуск і зупинку системи, вибір режиму роботи, оперативне керування, відображення стану процесу. Усі клавіші повинні мати пряму або умовну позначку, а також індикацію включення.

Мова задач – це символи й побудовані з них фрази, використовувані для автоматичного керування по програмі. Символи мають певні значення (стандарт ISO 6983-1:2009), програма повинна мати назву, а також містити позначення початку й кінця. Структурно вона ділиться на глави (:) і кадри (N). Функції ЧПУ розділяються на стандартні, технологічні, геометричні й функції знаків.

Символи, букви й цифри, які утворюють конкретну функцію, називаються **терміналами** мови. Усі термінали діляться на шість класів:

1. Стандартні функції: G (основні), M (допоміжні для управління автоматикою).
2. Цифри: 0...9.
3. Технологічні функції: N – номер кадру, F – функція подачі, S – функція швидкості привода головного руху, T – функція позиції інструмента, % – початок програми, LF – кінець кадру).
4. Геометричні функції: [X, Y, Z] – первинні координати, [I, J, K] – координати початку руху, [P,Q] – вторинні координати, [U,V,W] – третинні координати, [A,B,C] – кутові координати.
5. Функції знака: + (плюс), – (мінус).
6. Функція параметра R, що використовується для спрощення обчислень при програмуванні.

Основним інформаційним блоком управляючої програми ЧПУ є кадр.

Структуру кадру визначає формат УП.

Формат УП – це умовний запис кадру з максимальним обсягом інформації, що визначає набір застосовуваних слів, порядок їх розташування та обсяг інформації кожного слова.

Існує зручний формалізм, який дозволяє описати формат кадру управляючої програми, прийнятий для конкретного пристрою ЧПУ. Цей формалізм, по-перше, вводить перелік використовуваних символів, а по-друге, вказує, як повинна бути побудована числова частина кожного слова.

Для пояснення правил розглянемо такий формальний запис, що розкриває *структуру і формат* деякої управляючої програми:

% : / N3 G2 X+053 Z+053 F031 S04 T04 M2

З наведеного прикладу витікає, що пристрій ЧПУ сприймає символ (%) початку програми, символ (:) головного кадру і символ (/) пропуску кадру, а також має таке *кодування* інформації:

N3 – тризначний номер кадру з ведучими нулями у слові;

G2 – двозначна підготовча функція з ведучими нулями у слові;

X+053 – восьмирозрядне переміщення по осі X із знаком, п'ять цифр до десяткової крапки і три після, ведучі нулі дозволено опускати;

F031 – швидкість подачі з трьома цифрами до десяткової крапки, одна після і ведучі нулі дозволено опускати;

T04 – чотиризначна функція інструмента, ведучі нулі можна опускати;

M2 – двозначна допоміжна функція.

У мануалі до конкретного пристрою ЧПУ, прикладеного до верстата, завжди описуються формат кадру та УП. Умовний запис формату УП показує, як необхідно формувати його при конкретному програмуванні для даного верстата.

У документації до верстата наводяться також наступні дані: перелік і призначення всіх реалізованих підготовчих і допоміжних функцій; таблиці кодів швидкостей подач і головного руху; таблиці кодів номерів позицій інструмента; перелік номерів коректорів із вказанням їх призначення та особливостей застосування; межі розмірних переміщень по всіх осях координат; перелік усіх сприйманих і реалізованих символів кодового набору; перелік і кодові номери всіх підпрограм, що зберігаються в пам'яті пристрою ЧПУ.

Зміст кадру УП також регламентований. Кожен кадр повинен містити: слово “Номер кадру”, інформаційні слова, символ “Кінець кадру”. В залежності від виконання верстата зміст кадру може розрізнятися, але послідовність запису символів повинна відповідати стандарту на формат кадру. При використанні символів табуляції вони проставляються перед кожним інфор-

маційним словом (за винятком “Номер кадру”) у кадрі УП.

Інформаційні слова в кадрі записуються в такій послідовності:

1. Підготовча функція (G).
2. Розмірні переміщення (X, Y, Z, U, V, W, P, Q, R, A, B, C).
3. Параметр інтерполяції або крок різьби (I, J, K).
4. Функція подачі (F).
5. Функція головного руху (S).
6. Функція інструмента (T).
7. Допоміжна функція (M).

В пристроях ЧПУ нового покоління десятковий код функції – тризначний. Слід зважити на те, що підготовчі функції розбиті на групи. У кадрі можна задавати тільки одну функцію з кожної групи. Підготовчі функції записуються в кадрі одне за одним у порядку зростання їх кодових номерів.

1.2 Варіанти архітектурної організації сучасних систем ЧПУ

Класифікація архітектурних розв'язків дозволяє простежити еволюцію ЧПУ, а також зрозуміти зміст їх структурних перетворень. Системи ЧПУ, які випускаються в цей час можна розділити на дві групи:

1. *Програмно-апаратні системи CNC* (англ. *Computer numerical control*). Ці класичні системи випускаються лише фірмами, які мають досвід розробки власної високоякісної мікроелектронної апаратури. У цей час фірми пропонують *модифікацію класичної системи з персональним комп'ютером*, який використовується як термінал для задоволення потреб користувачів, що бажають мати гнучкий інтерфейс оператора.

2. *Однокомп'ютерні й двухкомп'ютерні системи PCNC* (англ. *Personal Computer Numerical Control*). Ці системи містять дві компоненти – термінальну (PC) і прикладну (NC). Кожна з компонентів реалізована або на окремому комп'ютері (двухкомп'ютерний варіант), або на одному комп'ютері, у якому на окремій платі реалізоване ядро операційної системи реального часу (однокомп'ютерний варіант).

Варіанти реалізації відбивають сумарний досвід розроблювачів систем ЧПК й перспективні тенденції. Далі докладніше про названі системи.

Система CNC

Системи цього сімейства побудовані за принципом багатопроцесорних, які містять у собі ЧПУ-процесор, процесор програмувального контролера автоматизації й графічний процесор.

Система може бути також оснащена пасивним терміналом або промисловим комп'ютером з операційною системою Windows 98, як, наприклад, система NUM (Франція), структурна схема якої наведена на рисунку 1.1.

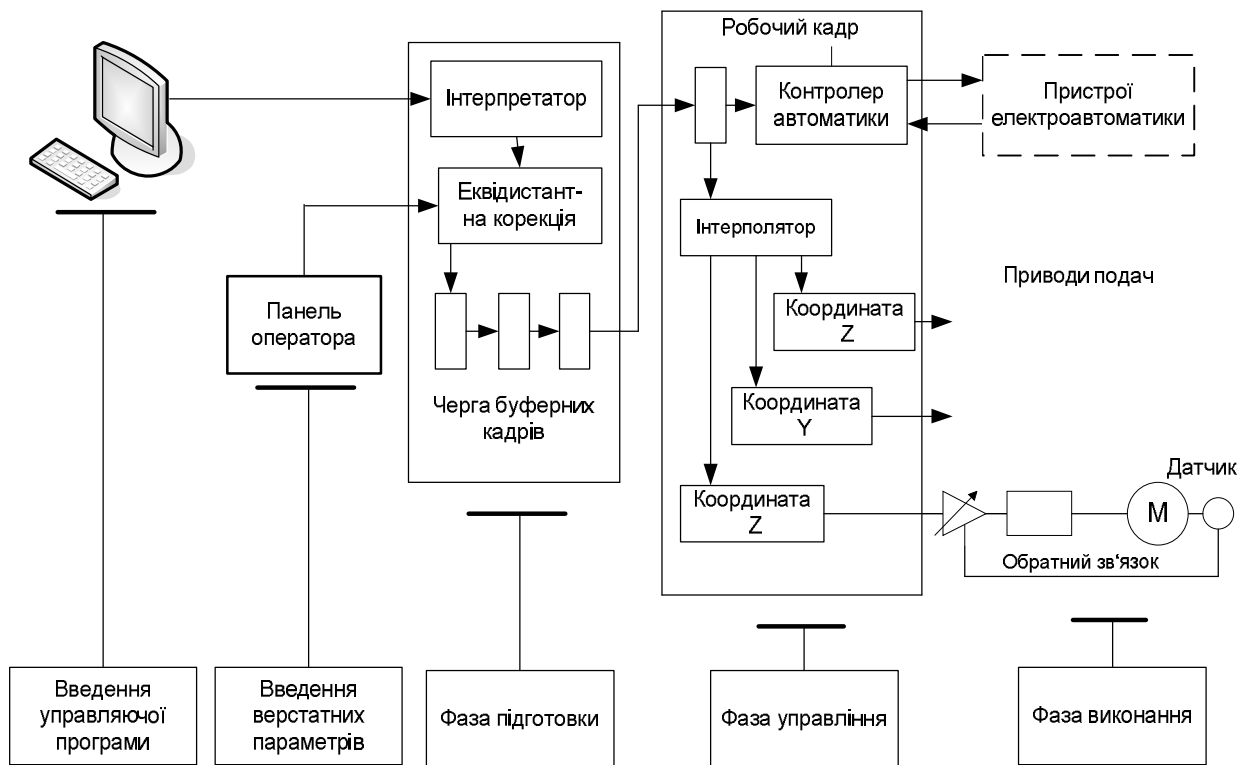


Рисунок 1.1 – Архітектура системи ЧПУ класу CNC фірми NUM

Пристрої ЧПУ цих систем мають компактні модулі й різняться числом координат, а також числом входів-виходів для підключення засобів електроавтоматики й мережних ресурсів. Ці системи дозволяють формувати незалежні канали ЧПУ з координатних груп, допускають використання як аналогових так і цифрових приводів, а також підключення до локальної мережі.

Висока обчислювальна потужність систем NUM забезпечує широкий набір їх функціональних можливостей. У них передбачені сплайновий і поліноміальний (до п'ятого порядку) алгоритми інтерполяції, п'ятикоординатна корекція інструмента, одночасна робота із двома різними управляючими програмами, 3D-графіка й інше.

У системах з термінальним комп'ютером можлива адаптація інтерфейсу оператора до запитів кінцевих користувачів, а також діалогове програмування за допомогою інструментальних систем PROGRAM_MILL і PROGRAM_TURN.

Системи PCNC

Фірма Allen Bradley у складі концерну Rockwell (США) випускає широке сімейство систем ЧПУ – від традиційної CNC-системи до систем PCNC з персональним комп'ютером. Система PCNC цієї фірми виконана за схемою, наведеною на рисунку 1.2.

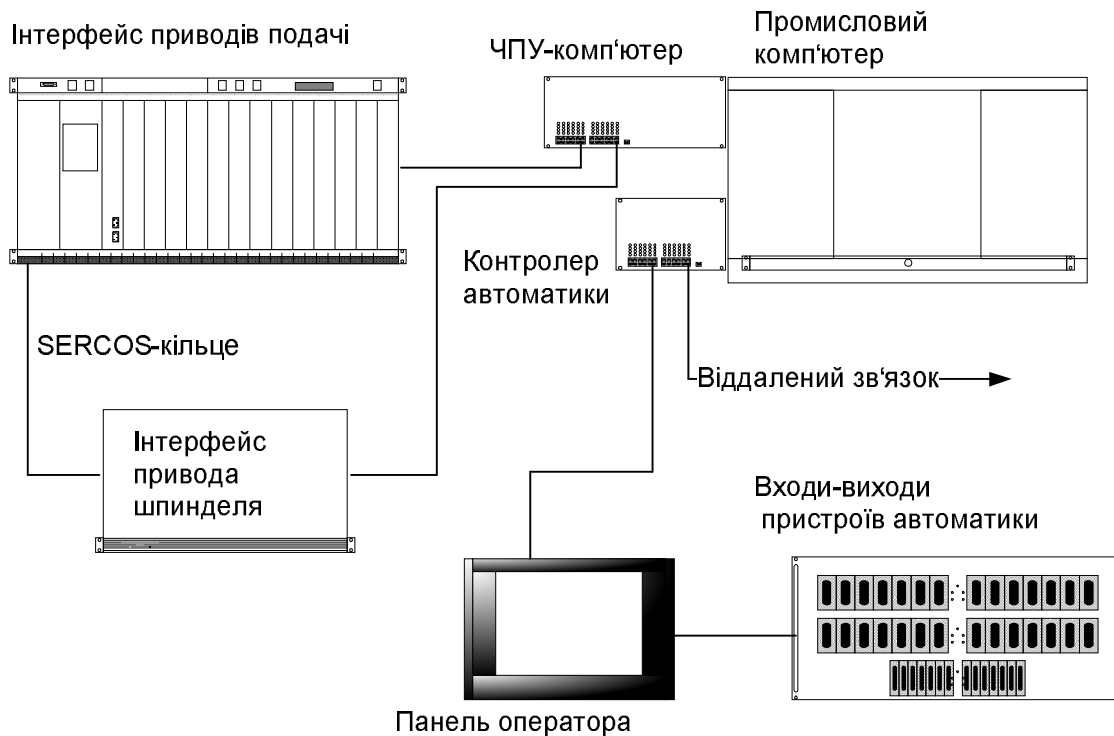


Рисунок 1.2 - Архітектура системи ЧПУ класу PCNC-1 фірми *Allen-Bradley*

Система містить у собі:

- спеціалізований промисловий комп'ютер з Windows NT операційною системою й можливістю розробляти користувацькі додатки на Visual Basic;
- PCI-одноплатний ЧПУ-комп'ютер, який виконує всі функції ядра, у тому числі програмно-реалізованого контролера електроавтоматики. PCI (англ. Peripheral component interconnect) – шина материнської плати комп'ютера, максимальна частота 533 Мб/с, 64 розрядів.

Програмування й редагування контролера здійснюються через загальний для всієї системи термінал. Контролер має власну мережу.

Система ЧПУ фірми ANDRON (Німеччина) ставиться до повного двужкомп'ютерного варіанту. Її структура представлена на рисунку 1.3.

Система містить у собі наступні модулі:

- модуль термінального комп'ютера;
- модуль ЧПУ-комп'ютера;
- панель оператора й монітор;
- модулі віддалених входів-виходів програмувального контролера;
- одну або кілька груп цифрових (SERCOS) приводів подачі й головного привода.

Апаратура системи практично повністю складається з покупних компонентів і плат. У зв'язку із цим фірма ANDRON не приховує деталей апаратної реалізації.

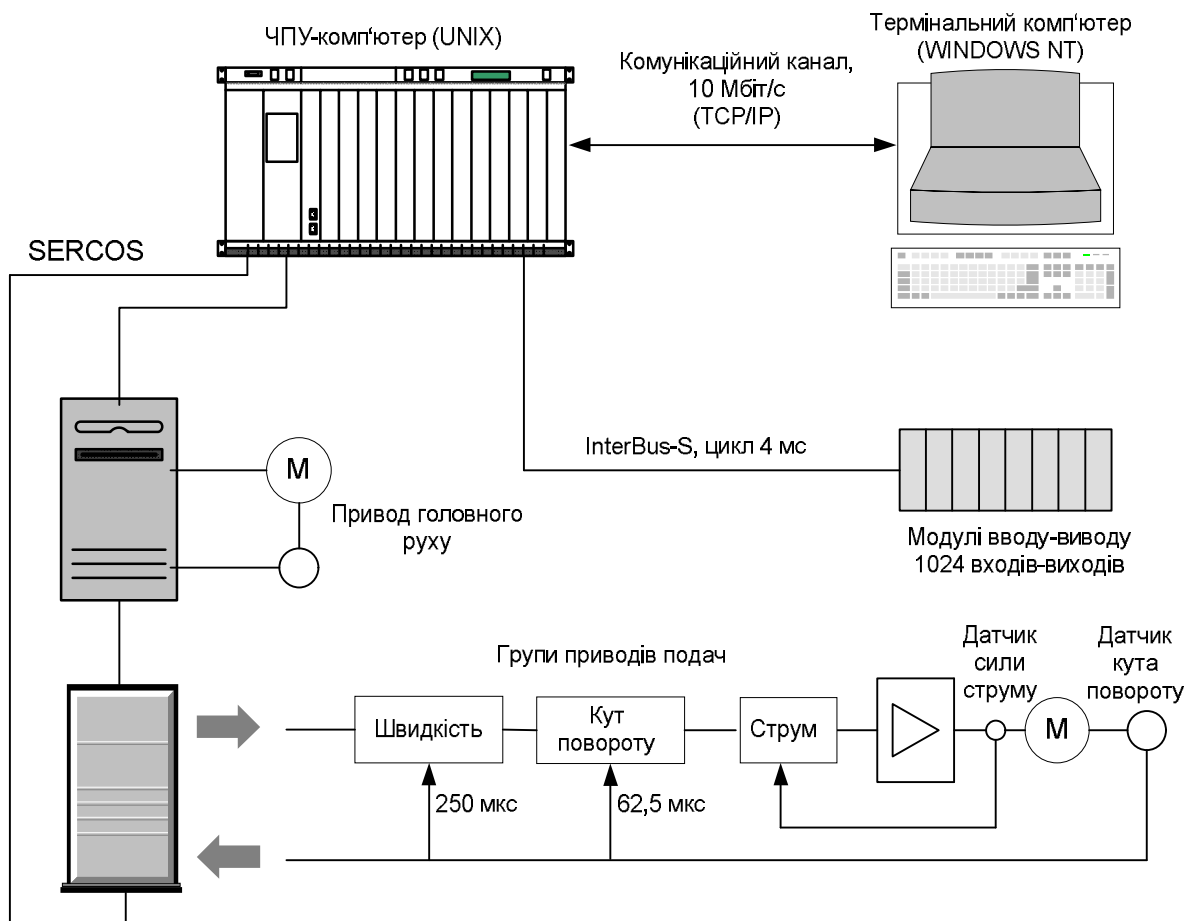


Рисунок 1.3 - Система PCNC з Sercos-інтерфейсом

До складу **термінального комп'ютера** входять:

- материнська плата з Celeron-процесором і інтегрованими контролерами SCSI, VGA, TFT, IDE;
- багатофункціональна інтерфейсна плата MFA з пам'яттю CMOS-ROM (трансп'ютерний контролер комунікаційного каналу, який зв'язує термінальний комп'ютер зі ЧПУ-комп'ютером).

Усі плати встановлені на пасивній ISA-шині. Передбачена установка додаткових резервних плат – внутрішнього модему, мережної плати, SCSI-плати. Для спеціальних завдань можлива установка PCI-плат.

До складу **ЧПУ-комп'ютера** входять:

- материнська плата з Celeron-процесором;
- плата MIO (Main Input-Output) підтримки комунікаційного інтерфейсу з термінальним комп'ютером (зі швидкістю 10 Мбіт/с), а також плата інтерфейсу маховичка ручного переміщення;
- плата програмувального контролера з інтерфейсом Interbus-S (із циклом 4 мс для 1024 входів-виходів);
- одна або кілька плат Sercos-інтерфейсу з мікросхемою SERCON 410-B.

Кожний SERCOS-інтерфейс обслуговує з періодичністю 0,5 мс одну із трьох груп автономних приводів подачі й привод шпинделя. Приводи однієї групи включені в кільцеву оптоволоконну мережу.

У платформі системи ЧПУ фірми ANDRON апаратний рівень розташований під операційною системою Windows NT у термінальному комп'ютері й під оригінальною операційною системою реального часу (ОС РЧ) у ЧПУ-комп'ютері.

На прикладному рівні термінальний комп'ютер відкритий для різноманітних додатків і спеціальних діалогів кінцевого користувача. Ці додатки можна назвати САМ-додатками. Для побудови САМ-додатків передбачена інструментальна мова ANLOG-C, що забезпечує доступ до функцій ядра в ЧПУ-комп'ютері.

Система ЧПУ фірми Boschrexroth побудована по класичному двух-комп'ютерному варіанту (рис. 1.4).

Термінальний комп'ютер має операційну систему Windows NT, а ЧПУ-комп'ютер – операційну систему UNIX. Зв'язок операційних середовищ здійснюється за допомогою протоколів TCP/IP, які допускають вилучене розміщення терміналу й роботу декількох терміналів з одним ЧПУ-комп'ютером. У свою чергу, ЧПУ-комп'ютер допускає багатоканальну роботу більш ніж з однієї управляючою програмою.

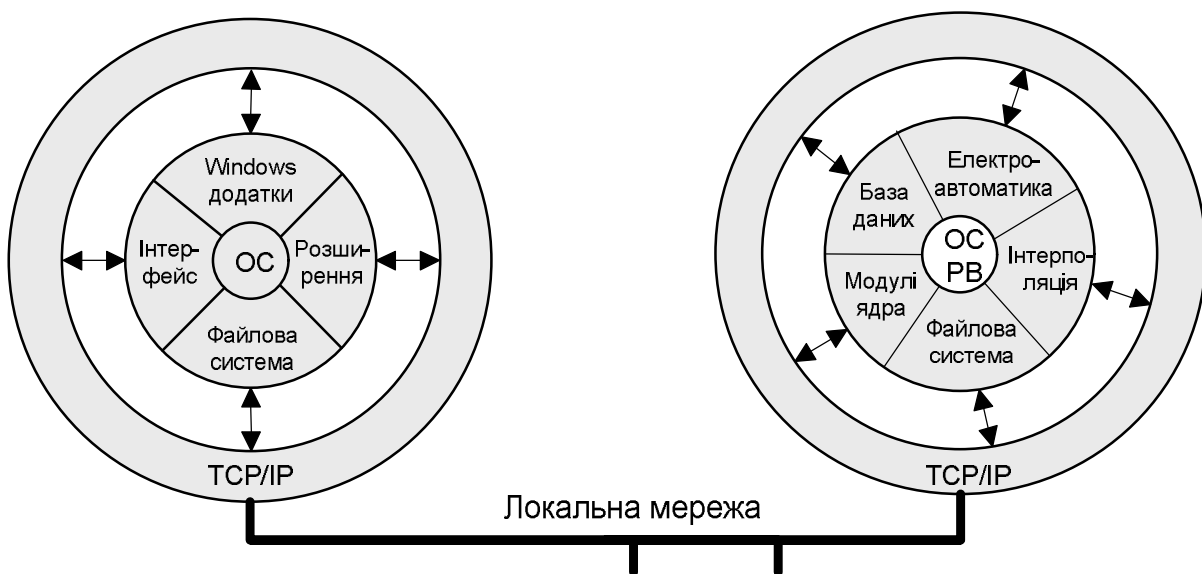


Рисунок 1.4 – Архітектура системи ЧПУ класу PCNC-2 фірми Boschrexroth

Прикладне математичне забезпечення термінального комп'ютера й прикладне математичне забезпечення ядра в ЧПУ-комп'ютері оточені оболонкою з декількох сотень інтерфейсних АРІ-функцій (Application Programming Interface), які забезпечують кінцевим користувачам можливість розробляти власні додатки й розширення.

В оболонку термінального комп'ютера включена потужна DLL-бібліотека NCS (Numerical Control System) класів об'єктів, «покриваюча» API-функції, яка спрощує розробку нових додатків.

Фірма *Deltatau* (Великобританія) розробила двухкомп'ютерний варіант PCNC, у яким ЧПУ-комп'ютер виконаний у вигляді окремої плати РМАС (Programmable Multi-Axes Controller), установленюваної на шину ISA або PCI термінального персонального комп'ютера (рис. 1.5).

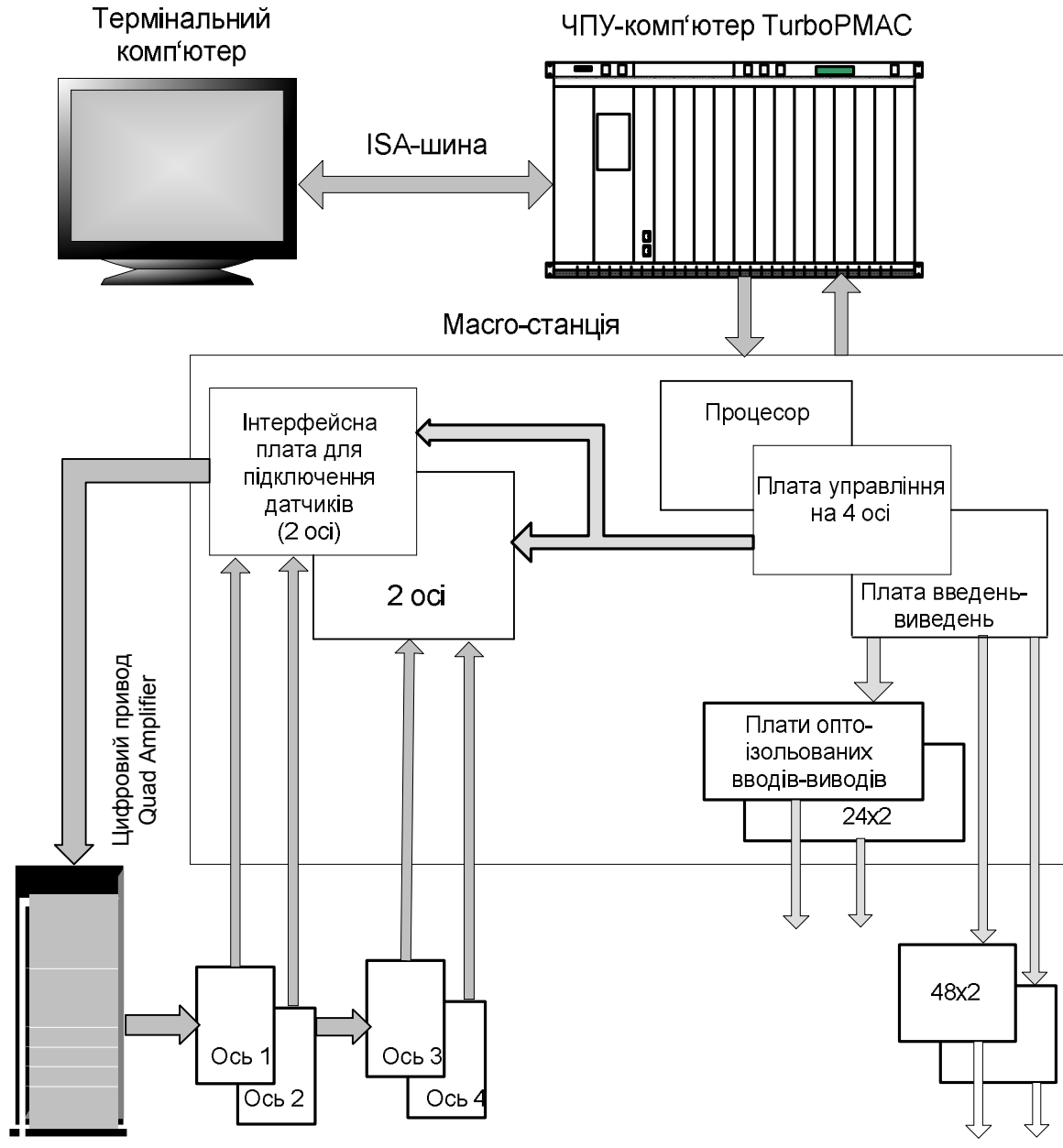


Рисунок 1.5 – Архітектура системи ЧПУ класу PCNC фірми *Deltatau*

Термінальний комп'ютер з Windows NT операційною системою виконує класичні функції термінального завдання й функції інтерпретатора управляючих програм.

Одноплатний ЧПУ-комп'ютер РМАС на процесорі Motorola 56300 ви-

рішує геометричні й логічні завдання, виконуючи функції інтерполятора, контролера приводів подачі й шпинделя, а також програмно-реалізованого контролера електроавтоматики.

Інтерполятор забезпечує всі види інтерполяції, включаючи сплайнову, розгін і гальмування, випереджаючий перегляд кадрів Look Ahead, а також циклічне формування управляючих впливів з періодом 440 мкс. Контролер електроавтоматики, реалізований програмно, працює у фоновому режимі й підтримує паралельне керування 64 циклами електроавтоматики.

Контролер приводів здатний управляти 32 координатними осями, згрупованими в 16 координатних систем. Він ухвалює сигнали позиційних датчиків зворотного зв'язку, замикає позиційні контури, виконує функції ПД-регулятора, імітує в цифровому виді сигнали зворотного зв'язку по швидкості, виробляє в цифровому виді широтно-імпульсний сигнал для приводів подачі й аналоговий сигнал ± 10 В для привода головного руху.

Вхідні сигнали для управління приводами й електроавтоматикою надходять у кільцевий оптоволоконний канал, який забезпечує дистанційне управління об'єктами зі швидкістю передачі даних 125 Мбіт/с. Приймаючим пристроєм служить інтелектуальний периферійний термінал Macro-станції (Motion and Control Ring Optical). У кільце можна включити кілька таких терміналів.

Термінал замикає швидкісні контури восьми приводів. Він приймає в блоках ACS (Axes Coordinate System) сигнали обмежників робочої зони й датчиків нульових точок координатних систем і формує сигнали управління двигунами будь-якого типу (асинхронними, постійного струму й ін.). Блок Quad Amplifier забезпечує управління чотирма двигунами загальною потужністю до 25 кВт. Інша функція периферійного терміналу - управління електроавтоматикою через модулі оптоізованих входів-виходів.

Набір модулів ЧПУ-комп'ютера РМАС і станції Macro орієнтований на побудову власних систем ЧПУ у кінцевих користувачів шляхом розробки оригінального термінального завдання й інтерпретатора в середовищі промислового персонального комп'ютера. Однак самі модулі є для кінцевого користувача «чорними ящиками» і їх архітектура закрита.

Таким чином, в архітектурних розв'язках лідируючі позиції займає концепція PCNC. При цьому в міру росту обчислювальної потужності процесорів перевага віддається однокомп'ютерному варіанту.

Як операційна система стандартом де-факто прийнята система Windows NT з розширенням реального часу. Програмувальні контролери реалізуються програмним шляхом у рамках єдиного обчислювального середовища для ядра ЧПУ, а термінал системи ЧПУ використовують для програмування електроавтоматики.

Периферія систем ЧПУ стає мережною, причому все частіше єдина ме-

режа використовується як для приводів подачі, так і для системи керування електроавтоматикою. Спостерігається тенденція в розвитку ідей відкритої архітектури, яка надає кінцевому користувачеві широкі можливості для реалізації власних функцій.

1.3 Варіанти реалізації відкритої архітектури систем ЧПУ

Гнучкі й найбільш складні системи ЧПУ з відкритою архітектурою виконують згідно двухкомп'ютерної архітектурної моделі (рис. 1.6).

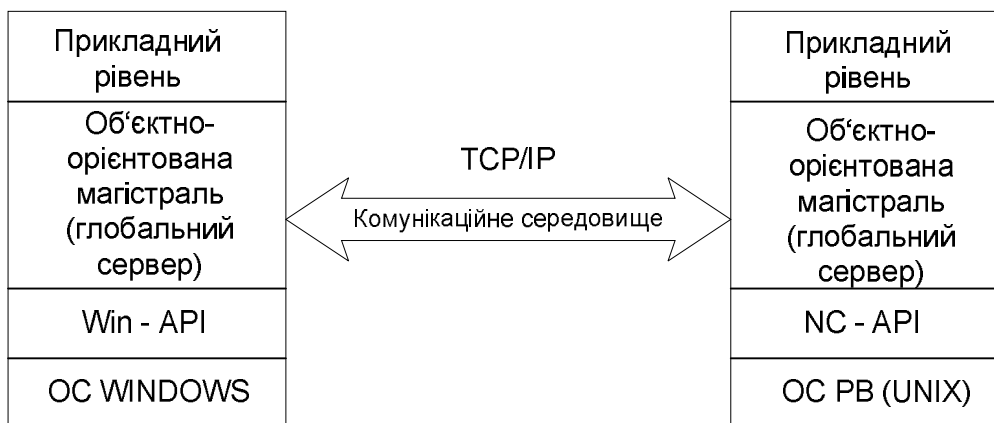


Рисунок 1.6 – Двухкомп'ютерна архітектурна модель системи ЧПУ

Двухкомп'ютерна модель передбачає розміщення РС-підсистеми на одному комп'ютері, а NC-підсистеми – на іншому.

У РС-підсистемі найбільш доцільна операційна система Windows NT, а в NC-підсистемі – операційна система реального часу, наприклад, UNIX. Обидві операційні системи сумісні в тому розумінні, що підтримують комунікаційні протоколи TCP/IP. Це дозволяє побудувати комунікаційне середовище, яке поєднує обидві підсистеми. Включення в це середовище прикладного рівня з функціями доступу до інтерфейсів модулів (а загальне число таких функцій може досягати декількох сотень) створює віртуальну шину, яка забезпечує низько рівневі послуги доступу. Об'єктну надбудову в шині формує глобальний сервер, тобто єдина для обох підсистем *об'єктно-орієнтована магістраль*.

Однокомп'ютерна модель допускає використання комп'ютера, оснащеного додатковими *контролерами* для зв'язку з мехатронними об'єктами управління (рис. 1.7). У їхньому числі можуть бути контролер стежачи приводів, програмувальний контролер PLC (Programmable Logic Controller), спеціальні пристрої для управління технологічними процесами й ін.

Як операційна система може бути використана система Windows NT, постачена відповідним розширенням, наприклад, у вигляді системи RTX 4.1 американської фірми Venturecom.

Слід зважити на те, що Windows NT не може використовуватися в ре-

жимі реального часу по наступних причинах:

- недостатня кількість real-time пріоритетів;
- відсутність спадкування пріоритетів, як засобу боротьби з інверсією пріоритетів;
- не підходяща для RTOS (операційних систем реального часу) *система обробки переривань*.

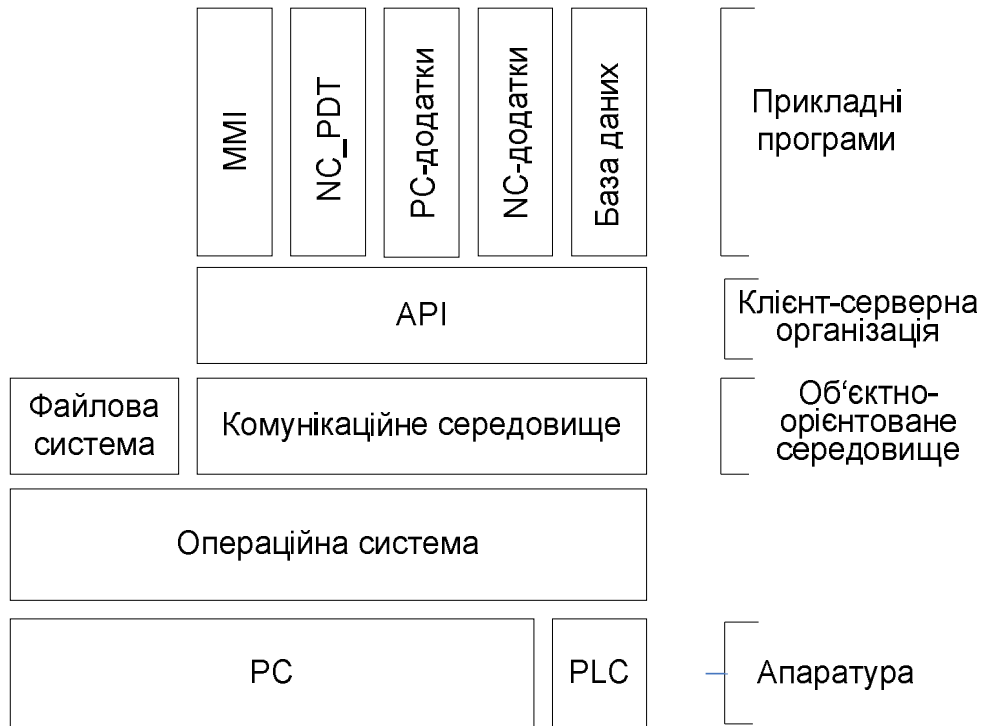


Рисунок 1.7 - Однокомп'ютерна архітектурна модель системи ЧПУ

В Windows NT доступ до переривань здійснюється із драйвера ядра, а самі переривання обробляються у два етапи:

- на першому етапі викликається дуже коротка програма (Interrupt Service Routine, ISR), яка здійснює критичну обробку;
- на другому етапі відбувається основна обробка переривання в програмі Deferred Procedure Call (DPC).

Особливістю обробки переривань в Windows NT є те, що всі обробки переривань (DPC) виконуються з однаковим рівнем пріоритету в порядку вступу (принцип FIFO). При цьому час закінчення обробки DPC виявляється залежним від непередбаченої активності інших драйверів системи. Це неприпустимо для систем реального часу, які будуються за принципом жорсткого дедермінізму – необхідно точно знати максимальний час від моменту виникнення переривання до входу в процедуру обробки з гарантією, що цей час не буде перевищений.

Фірма Venturecom, будучи партнером Microsoft, одержала право вста-

новлювати свій вихідний код у шар HAL (Hardware Abstraction Layer) операційної системи Windows NT. Фірма Venturesom розробила *систему RTX* (Real Time extention), яка модифікує шар HAL і доповнює його диспетчером потоків (threads) реального часу. Цей диспетчер ізолює переривання, що створює можливість будувати додатки реального часу, про існування яких будь-які інші додатки нічого не знають.

Підсистема реального часу RTSS (Real-Time Sub-System), яка реалізована у вигляді драйвера Windows NT, служить доповненням до операційної системи й використовує сервіси Windows NT і HAL для роботи додатків реального часу окремо від будь-яких інших додатків. Виконуючи власні функції, ця підсистема здійснює керування ресурсами RTX.

Після установки RTX стандартна NT перетворюється в операційну систему реального часу із жорстким детермінізмом (hard real-time). При цьому сама NT про це не підозрює, тому що ні ядро підсистеми, ні виконавчий блок NT не змінені. Підсистема реального часу видна з Windows NT, як ще один *драйвер* пристрою.

На інтерфейсному рівні прикладні програмні інтерфейси Win32 і RTX схожі. У них реалізовані функції, необхідні для створення звичайних додатків (Win32) і додатків реального часу (RTX). При цьому розроблену з використанням RTX програму можна налагоджувати й запускати в середовищі Win32.

Архітектурні варіанти, представлені на рисунках 1.6-1.7, розроблені з урахуванням *принципів відкритої архітектури* відносно ЧПУ.

Принципи відкритої архітектури зводяться до наступного:

- чітке розмежування між системними, прикладними і комунікаційними компонентами;
- можливість незалежного розвитку кожного із цих компонентів як на основі оригінальних розробок, так і шляхом вбудовування покупних програмних систем;
- клієнт-серверна організація взаємодії підсистем;
- стандартизація інтерфейсів і транзакцій.

У *вертикальному перерізі PC-підсистема* має багаторівневу структуру (рис. 1.8) і повною мірою відповідає моделі віртуальної машини.

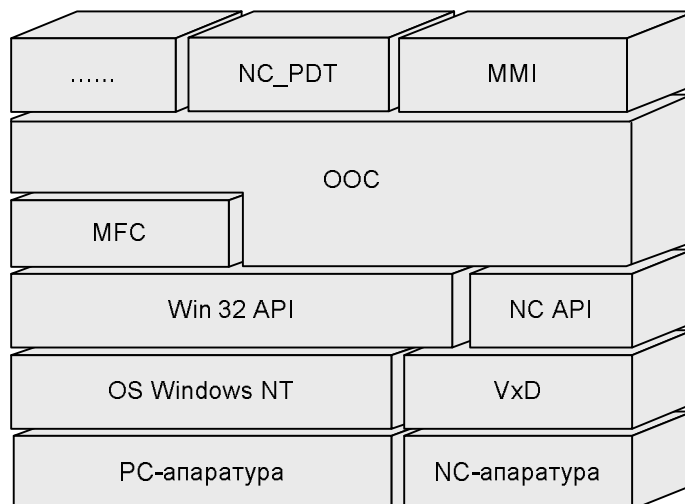


Рисунок 1.8 - Віртуальна модель PC-підсистеми

Нижній рівень представляє апаратура – комп'ютер і контролер (або контролери). Вище розміщується операційна система Windows NT разом із драйверами віртуальних пристроїв (VxD), які забезпечують керування зовнішніми пристроями.

Доступ до операційної системи і її службам здійснюється за допомогою API-шару (прикладний інтерфейс), який підтримується Win32-функціями й NC-функціями. Зазначені функції забезпечують вхід у підсистеми Windows NT і NC. Функції реалізовані у вигляді DLL (Dynamic Link Library, бібліотек з динамічним зв'язуванням).

Вище API-шару розташований об'єктно-орієнтований сервер, який створює фундамент для додатків у системі PCNC.

Об'єктно-орієнтований сервер містить у собі об'єкти стандартних класів з бібліотеки MFC (Microsoft Foundation Classes), а також спеціально розроблені класи OOC_CL об'єктно-орієнтованої магістралі OOC (Object Oriented Channel). Об'єктно-орієнтований сервер містить загальні для всіх додатків алгоритми – оброблювачі помилок, засоби форматування й конвертування даних, управляючі елементи багатовіконного екрана й ін.

На прикладному рівні розміщуються різноманітні додатки: інтерфейс користувача MMI (Man Machine Interface), інструмент розробки й верифікації керуючих програм NC_PDT (NC Program Data Tool) і ін.

Слід урахувати, що в *однокомп'ютерному варіанті* розподіл роботи допускає оптимальне використання обчислювальних ресурсів системи для реалізації необхідного масштабу реального часу.

З обліком викладеного можна зробити наступні висновки:

- Архітектура системи ЧПУ визначається кількістю й складом **завдань управління**.
- Архітектурне компонування системи ЧПУ являє собою сукупність

модулів. Кожний модуль автономний і є вкладеним об'єктом. Він має власну структуру даних і алгоритмів, а також власну інтерфейсну оболонку для роботи в клієнт-серверному середовищі.

- У двухкомп'ютерній архітектурі комунікаційне середовище, яке поєднує дві операційні системи, підтримується протоколами TCP/IP.
- В однокомп'ютерній архітектурі використовується операційна система Windows NT з розширенням реального часу. Як комунікаційне середовище використовується об'єктно-орієнтована магістраль, яка реалізує функції сервера.

1.4 Організація зв'язків між компонентами системи керування

Паралельний інтерфейс. У стандартну комплектацію PC входить 8-розрядний паралельний інтерфейс Centronics, який призначений для однієї передачі інформації від PC до периферійного встаткування на відстань до 2 м. зі швидкістю 100 Кбайт/с.

Фірми Intel, Xircon, Zenith і інші спільно розробили специфікацію поліпшеного порту EPP (Enhanced Parallel Port), який дозволяє здійснювати двунправлену передачу по каналу прямого доступу. Для використання такого порту необхідно спеціальне програмне забезпечення. EPP забезпечує швидкість обміну даними до 2 Мбіт/с і підключення в ланцюжок до 64 периферійних пристроїв. Розроблений також порт ECP (Extended Capability Port), який дозволяє підключати до 128 пристроїв.

Послідовний інтерфейс. У стандартну комплектацію PC входить найпоширеніший інтерфейс RS-232C, відомий як CCITT V.24. У промисловості найбільше часто застосовують інтерфейс RS-485, який використовує симетричну двухпроводну лінію зв'язку й дозволяє будувати мережі із числом абонентів до 32 на відстань до 1200 м.

Найпоширенішим мережним розв'язком є мережа Ethernet зі швидкістю до 100 Мбіт/с. Однак протокол Ethernet має недетерміновану природу, яка суперечить вимогам керування в реальному масштабі часу. Тому для забезпечення гарантії доставки повідомлень у заданий інтервал часу застосовують поліпшений протокол ATM (Asynchronous Transfer Mode) зі швидкістю обміну 25 Мбіт/с або мережу PROFINET, розроблену фірмою SIEMENS на базі мереж Ethernet і PROFIBUS.

В усі центральні процесори блоків керування SIMATIC S7 вбудований MPI інтерфейс. Він може бути використаний для побудови простих і найбільш дешевих мережних структур. MPI інтерфейс дозволяє підтримувати одночасний зв'язок одного блоку керування SIMATIC S7 з декількома програматорами або комп'ютерами, пристроями й системами людино-машинного інтерфейсу, програмувальними контролерами SIMATIC S7-300/400, блоками

керування SIMATIC S7, системами комп'ютерного керування SIMATIC WinAC. При цьому обмін даними може здійснюватися з 16 центральними процесорами (з використанням STEP 7 V4.x).

MPI інтерфейс блоків керування SIMATIC S7 безпосередньо пов'язаний із внутрішньої комунікаційної K-шиною центрального процесора, яка забезпечує доступ через MPI інтерфейс до всіх модулів, підключених до K-шини. Швидкість передачі даних через MPI інтерфейс – 187,5 Кбіт/с. Комунікаційні компоненти для MPI зв'язку – це мережні кабелі, з'єднувачі й повторювачі RS 485 зі спектра компонентів PROFIBUS.

Зв'язок між PC і контролерами. Для зв'язку між PC і контролерами прийнято застосовувати послідовні шини. До цієї групи ставляться кілька європейських розробок: PROFIBUS (DIN 19245), PROFINET, Bitbus, CAN, Interbus S, а також американська Fieldbus HART.

Зв'язок між контролерами й приводами. Зв'язок із слідкуючими приводами залежить від типу привода. При використанні цифрових приводів подачі фірм INDRAMAT і Bosh зв'язок може бути організований за допомогою інтерфейсу SERCOS (Serial Real-time Communication System) – по стандарту IEC 61491. Система SERCOS являє собою кільцеву оптоволоконну мережу, вузлами якої є програмно-апаратні модулі. Такий модуль складається зі спеціального однокристалного контролера й трансиверної (приймально-передавальної) частини, причому провідний модуль може бути оформлений у вигляді плати, встановлюваної в PC. Крім одного провідного, усі інші модулі є веденими. Комунікаційна сесія здійснюється циклічно з постійною частотою, що залежить від числа ведених модулів у мережі, періодичність циклів настраюється на етапі ініціалізації системи. Так, управляти п'ятьма слідкуючими приводами можна з періодом 1 мс, а вісьма приводами (максимальне число) з періодом 2 мс. У кожному циклі задаються швидкість і крутний момент, від кожного привода збирається інформація про фактичні значення цих параметрів. Обмежень на загальне число одночасно працюючих приводів практично не існує. Довжина міжвузлового сегмента для пластикових оптоволоконних кабелів може досягати 60 м, а для скляних 250 м.

Найпоширенішим протоколом зв'язку на рівні керування встаткуванням з апаратурою фірми SIEMENS залишається PROFIBUS. Для приводних систем SINAMICS фірмою Siemens розроблений інтерфейс DRIVE-CLIQ, який спрощує вирішення комунікаційних завдань із енкадерами (інкрементальними та абсолютними датчиками переміщень).

Зростає визнання інтерфейсу CAN (Controller Area Network), розробленого фірмою Bosh. Він являє собою послідовний інтерфейс, спеціально створений для з'єднання між собою датчиків, виконавчих пристроїв і інтелектуальних контролерів. Переваги інтерфейсу CAN – забезпечення режиму обміну в реальному масштабі часу завдяки можливості *ініціативної передачі даних при зміні стану вхідних сигналів*, висока завадо стійкість і протокол з корекцією помилок. Інтерфейс CAN підтримує комунікаційні протоколи прикладно-

го рівня Devicenet фірми Allen-Bradly і CANopen, передбачає корекцію помилок, інтерфейс стійкий до промислових перешкод. Швидкість передачі через інтерфейс CAN – 1 Мбіт/с.

1.5 Особливості реалізації стандартів у системах ЧПУ

Основним стандартом технології взаємодії між додатками в архітектурі клієнт-сервер є стандарт OPC (англ. OLE for Process Control).

В основу технологія взаємодії *розподілених* додатків були покладені три базові принципи:

1. Незалежність від фізичного розміщення об'єктів. Компоненти програмного забезпечення не зобов'язані перебувати в одному виконавчому файлі, виконуватися в рамках одного процесу або розміщатися на одній апаратній системі.
2. Незалежність від платформи. Компоненти можуть виконуватися на різних апаратних і операційних платформах.
3. Незалежність від мови програмування. Відмінність у мовах, використаних для створення компонентів, не повинна перешкоджати їхній взаємодії.

Ідея стандарту OPC виникла на основі еволюції канонічної моделі архітектури відкритих систем ISO-OSI, представленої на рисунку 1.9.

7a: ISO MMS-додаток	7a: ISO MMS-додаток	MMS-подібний протокол над сокетами	MMS із клієнт- серверною взаємодією	Об'єктно- орієнтований MMS. ASNI-IDL перекодування
7b: ISO ACSE	7b: ISO ACSE		ASNI-BER перекодування	Метод віддаленого виклику через об'єктну шину
6: ISO	6: ISO		Виклик віддалених процедур RPC	
5: ISO	5: ISO			
4: ISO	Рівень адаптований для RFC 1006	Рівні TSP/IP	Рівні TSP/IP	Рівні TSP/IP (або інші)
3: ISO	Рівень TSP/IP			
Рівні 1 и 2 ISO	Рівні 1 и 2 ISO	Рівні 1 и 2 ISO	Рівні 1 и 2 ISO	Рівні 1 и 2 ISO
а	б	в	г	д

Рисунок 1.9 - Канонічна модель ISO-OSI і її еволюція

На рисунку 1.9, а стандартний додаток MMS (Manufacturing Message Specification, ISO 1090) займає прикладний рівень – рівень 7. Для організації зв'язку між додатками на цьому рівні використовується метод ACSE (Association Control Service Element), що дозволяє перевірити ідентичність і контекст додатків.

Проблема використання цієї моделі полягає в тому, що якщо буде потрібно створити розподілену систему керування, то така система перетворить-

ся в сукупність взаємодіючих віртуальних пристроїв – VMD (Virtual Manufacturing Device). При цьому стек протоколів VMD, який охоплює всі інші рівні ISO-OSI, стане занадто складним.

На рисунку 1.9,б показаний розв'язок, який використовує на третьому рівні ISO-OSI протокол TCP/IP (Transmission Control Protocol/Internet Protocol) і технологію міжмережної взаємодії, тобто технологію Internet. Як емулятор ISO-OSI послуг над протоколами TCP/IP на рівнях 2 і 3 застосовується *система вилучених запитів RFC 1006* (Remote Function Call).

На рисунку 1.9,в система MMS-послуг вибудована безпосередньо над сокетами TCP/IP. Сокет (socket) – це об'єкт, який є кінцевим елементом з'єднання, що забезпечує взаємодію між процесами транспортного рівня мережі. Перехідний варіант показано на рисунку 1.9, г. Тут використана *система дистанційного виклику RPC* (Remote Procedure Call).

І, зрештою, порівняно стійкий варіант розв'язку проблеми показано на рисунку 1.9, д. Тут протоколи TCP/IP у принципі необов'язкові.

Відкрите керування допускає максимальне використання *стандартів*, як мережних, так і на рівні кожної окремої системи керування. При цьому велика роль належить інтерфейсним стандартам, оскільки саме вони дозволяють побудувати відкриті розподілені системи керування, такі, як COM/DCOM і OPC. Для здійснення інформаційного обміну в системах ЧПУ використовується ряд специфічних стандартів, які охоплюють усі рівні – від терміналу до виконавчих пристроїв.

Ціль стандартів – забезпечити загальну роботу й взаємозамінність промислових пристроїв від різних виробників. Маючи затверджений у стандарті набір інтерфейсів, кінцевий користувач може організувати взаємодію й обмін даними між будь-якими розподіленими компонентами системи.

У системах ЧПУ застосовуються наступні стандарти.

- ***Стандарти термінальних функцій***

Тут застосовуються стандарти інформаційного обміну із зовнішніми пристроями: DXF, STEP (AP203, AP213, AP214, AP224), ISO 10303-41, ASCII, OPC, DDE. У мережних комунікаціях використовуються стандарти: TCP/IP, OSI/Ethernet, стандартні додатки MMS (в NC і PLC).

- ***Стандарти в логічному керуванні ЧПУ***

У розв'язку логічного завдання керування застосовуються стандарти IEC 6133-3 (мови програмування логічних контролерів), а для передачі сигналів керування виконавчими пристроями – RS-267C.

- ***Комунікація на рівні апаратури (BUS, VME, ISA, PCI).***

Уведення-вивід дискретних сигналів здійснюється згідно з вимогами стандартів RS-232, RS-485, SERCOS, ISA fieldbus, ISO 4336: «Числове програмне керування верстатів. При створенні комунікацій враховуються технічні вимоги до сигналів взаємозв'язку між пристроями числового керування й

електроустаткуванням верстатів із числовим керуванням», OBIOS (об'єктно-орієнтована система входів-виходів), а також стандарти OPC-SP95, IEEE P 145-1.2, DIM 1245, Profibus, Interbus, Devisenet.

1.7 Сутність виробничих стандартів STEP

В автоматизованих виробництвах ті види інтеграції, які відносять до області обробки на верстатах із ЧПУ, набувають найбільшого значення. Інтеграція цих систем здійснюється із застосуванням єдиної інформаційної моделі виробу в рамках його життєвого циклу – від комп'ютерного проектування й комп'ютерного планування до автоматизованої підготовки керуючих програм і виготовлення виробу на верстатах із ЧПУ. Подібна модель визначена рамками *стандартів STEP (Standard for the Exchange of Product model data)*.

У життєвому циклі виробу передбачені наступні фази:

- STEP-проектування CAD (Computer-Aided Design);
- макропланування технологічного процесу CAPP (Computer-Aided Process Planning);
- мікропланування операцій CAM (Computer-Aided Manufacturing);
- керування виготовленням NC (Numerical Control), які існують сьогодні поза STEP.

Фаза проектування

Фаза проектування (розробки) містить у собі генерацію й збереження STEP-даних для наступного виготовлення виробів.

У рамках фази розроблено кілька варіантів прикладних протоколів AP (Application Protocol), визначених як міжнародні стандарти. Кращим з них є протокол AP224.

Сутність протоколу AP224 полягає в наступному. На рівні фази розробки формується деякий повний набір інформації для планування (у наступній фазі) технологічних маршрутів. Цей набір містить у собі визначення даних у термінах 3D-геометрії (прямі, дуги і т.д.) і в технологічних термінах (кишеня, канавка, отвір, закруглення й ін.). Повноту набору забезпечує визначення розмірів і допусків, асоційованих з 3D-образом, а також генерація іншої важливої інформації, наприклад, відомості про матеріал, шорсткість, параметри закруглення гострих крайок і т.п.

Усі специфікації являють собою частину моделі, причому під протоколом AP224 розуміють і модель, і транслятор, який генерує у форматі AP224 виробничі дані не тільки для окремих деталей, але й для з'єднань. У складі транслятора є система керування базою даних (СКБД). Проект також може бути виконаний в САД-системі.

Фаза макропланування

У наступній фазі макропланування виробничі дані конструкторського

проекту обробляють у форматі AP213 у формі *технологічного маршруту* для верстатів зі ЧПУ.

Формат AP213 належить комплексу STEP, але поки ще не є міжнародним стандартом, а служить тільки входом у систему макропланування.

На рівні *макропланування* вирішуються наступні завдання:

- розробляється схема маршрутизації (здійснюється розподіл операцій маршруту по верстатах);
- складаються специфікації матеріалів, які узагальнюють маршрутну інформацію;
- формулюються вимоги до інструментального забезпечення;
- вибираються, замовляються або виготовляються необхідні пристосування й інструменти;
- визначаються норми часу для кожного переходу;
- складаються інструкції операторові.

На рівні макропланування система повинна прийняти інформацію у форматах STEP AP224 (оптимальний варіант) або в STEP AP203 (більш ранній варіант прикладного протоколу проектування), у форматі IGES (Initial Graphics Exchange) і інформацію креслення. Користувачеві повинна бути доступна модель оброблюваного виробу. У його розпорядженні повинні бути безліч екранів з інформацією про виріб, цехові ресурси й плани обробки. При цьому модель цехових ресурсів повинна включати опис верстатів, наявні матеріали й інструменти, оцінку часу обробки, технологічні можливості.

У процесі макропланування можна також оцінити вартість обробки.

Фаза мікропланування

Мікроплан – це кроки операцій, здійснювані числовим програмним керуванням. Він містить креслення й управляючі програми для верстатів із ЧПУ.

Стратегія фази мікропланування полягає в тому, щоб прийняти інформацію у форматі AP213, вибудувати як стабільна модель для стандартного входу в САМ-систему. САМ-система виконує мікропланування у форматі AP238 на основі стандарту STEP-NC для кожного верстата з тих, які визначені маршрутом операцій. При цьому САМ-система проектує траєкторії інструментів і з використанням постпроцесора переводять їх у формат коду ISO-7bit так, щоб забезпечити сумісність із конкретною системою ЧПУ. Крім того, розробляються схеми установок і корекції інструментів, а також докладні інструкції операторові.

Фаза виготовлення

Фаза виготовлення, як перехід САМ-NC, поки не реалізована. Для прямого використання інструкцій STEP AP238 повинні бути розроблені системи ЧПУ чергового покоління, які зможуть сприймати формат STEP-NC замість мови ISO-7bit (ISO 6983).

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ МОДЕЛІ

2.1 Керування процесами в операційній системі ЧПУ

Стани процесів

Найважливішою частиною операційної системи є підсистема керування процесами. Процес (або по-іншому, завдання) – це абстракція, що описує програму, яка виконується. Для операційної системи процес являє собою одиницю роботи, заявку на споживання системних ресурсів. Підсистема керування процесами планує виконання процесів, тобто розподіляє процесорний час між декількома одночасно існуючими в системі процесами, а також займається створенням і знищенням процесів, забезпечує процеси необхідними системними ресурсами, підтримує взаємодія між процесами.

У багатозадачній (багатопроектній) системі процес може перебувати в одному із трьох основних станів:

ВИКОНАННЯ – активний стан процесу, під час якого процес має всі необхідні ресурси й безпосередньо виконується процесором;

ОЧІКУВАННЯ – пасивний стан процесу, процес заблокований, він не може виконуватися по своїх *внутрішніх* причинах, він чекає здійснення деякої події, наприклад, завершення операції введення-виводу, одержання повідомлення від іншого процесу, звільнення якого-небудь необхідного йому ресурсу;

ГОТОВНІСТЬ – також пасивний стан процесу, але в цьому випадку процес заблокований у зв'язку із *зовнішніми* стосовно нього обставинами: процес має всі необхідні для нього ресурси, він готовий виконуватися, однак процесор зайнятий виконанням іншого процесу.

У ході життєвого циклу кожний процес переходить із одного стану в інший відповідно до *алгоритму планування процесів*, реалізованим у даній операційній системі. Типовий граф станів процесу показано на рисунку 2.1.

У стані **ВИКОНАННЯ** в однопроцесорній системі може перебувати тільки один процес, а в кожному зі станів **ОЧІКУВАННЯ** й **ГОТОВНІСТЬ** – кілька процесів. Ці процеси утворюють черги готових процесів.

Життєвий цикл процесу починається зі стану **ГОТОВНІСТЬ**, коли процес готовий до виконання й чекає своєї черги. При активізації процес переходить у стан **ВИКОНАННЯ** й перебуває в ньому доти, поки або він сам звільнить процесор, перейшовши в стан **ОЧІКУВАННЯ** якої-небудь події, або буде насильно "витиснутий" із процесора, наприклад, внаслідок вичерпання відведеного даному процесу кванта процесорного часу. В останньому випадку процес вертається в стан **ГОТОВНІСТЬ**. У цей ж стан процес переходить зі стану **ОЧІКУВАННЯ**, після того, як очікувана подія відбулася.

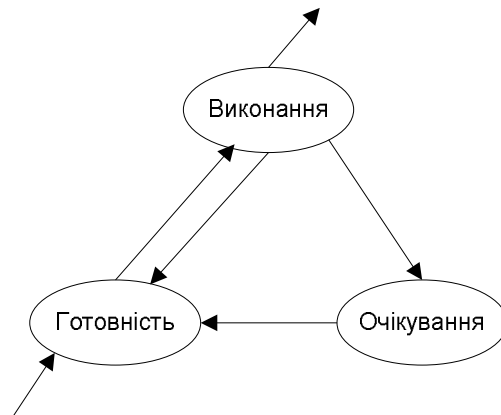


Рисунок 2.1 – Граф станів процесу в багатозадачному середовищі

Контекст і дескриптор процесу

Протягом існування процесу його виконання може бути багаторазово перерване й продовжене. Для того, щоб відновити виконання процесу, необхідно відновити стан його операційного середовища. Стан операційного середовища відображається станом реєстрів і програмного лічильника, режимом роботи процесора, показчиками на відкриті файли, інформацією про незавершені операції введення-виводу, кодами помилок виконуваних системних викликів і т.д. Ця інформація називається **контекстом процесу**.

Крім цього, операційній системі для реалізації планування процесів потрібна додаткова інформація: ідентифікатор процесу, стан процесу, дані про ступінь привілейованості процесу, місце знаходження кодового сегмента й інша інформація. У деяких ОС (наприклад, в ОС UNIX) інформацію такого роду, використовувану ОС для планування процесів, називають **дескриптором процесу**.

Дескриптор процесу в порівнянні з контекстом містить більш оперативну інформацію, яка повинна бути легко доступна підсистемі планування процесів. Контекст процесу містить менш актуальну інформацію й використовується операційною системою тільки після того, як ухвалено рішення про поновлення перерваного процесу.

Черги процесів являють собою дескриптори окремих процесів, *об'єднаних в списки*. Таким чином, кожний дескриптор, крім усього іншого, містить принаймні один показчик на інший дескриптор, що сусідить із ним у черзі. Така організація черг дозволяє легко їх переупорядковувати, включати й виключати процеси, переводити процеси з одного стану в інший.

Програмний код тільки тоді почне виконуватися, коли для нього операційною системою буде створений процес. Створити процес – це значить:

- створити інформаційні структури, що описують даний процес, тобто його дескриптор і контекст;
- включити дескриптор нового процесу в чергу готових процесів;

- завантажити кодовий сегмент процесу в оперативну пам'ять або в область свопінга (підкачування) для тимчасового збереження.

Алгоритми планування процесів

Планування процесів містить у собі розв'язок наступних завдань:

- визначення моменту часу для зміни виконуваного процесу;
- вибір процесу на виконання із черги готових процесів.

У системах ЧПУ для зміни процесів найчастіше застосовуються два типи алгоритмів – алгоритми, засновані на квантуванні, і алгоритми, засновані на пріоритетах.

Відповідно до алгоритмів, заснованих на квантуванні, зміна активного процесу відбувається, якщо:

- процес завершився й покинув систему,
- відбулася помилка;
- процес перейшов у стан ОЧІКУВАННЯ;
- *вичерпаний квант процесорного часу*, відведений даному процесу.

Процес, який вичерпав свій квант, переводиться в стан ГОТОВНІСТЬ і очікує, коли йому буде наданий новий квант процесорного часу, а на виконання відповідно до певного правила вибирається новий процес із черги готових. Таким чином, жоден процес не займає процесор надовго, тому квантування широко використовується в системах поділу часу. Граф станів процесу, зображений на рисунку 2.1, відповідає алгоритму планування, заснованому на квантуванні.

Кванти, виділювані процесам, можуть бути однаковими для всіх процесів або різними. Кванти, виділювані одному процесу, можуть бути фіксованої величини або змінюватися в різні періоди життя процесу. Процеси, які не повністю використовували виділений їм квант (наприклад, через відхід на виконання операцій уведення-виводу), можуть одержати або не одержати компенсацію у вигляді привілеїв при наступнім обслуговуванні. По різному може бути організована черга готових процесів: циклічно, за правилом "перший прийшов – перший обслужився" (правило FIFO) або за правилом "останній прийшов – перший обслужився" (правило LIFO).

Інша група алгоритмів використовує поняття "пріоритет" процесу. Пріоритет – це число, що характеризує ступінь привілейованості процесу при використанні ресурсів обчислювальної машини, зокрема, процесорного часу: чим вище пріоритет, тем вище привілеї.

Проблема синхронізації процесів

Процесам часто потрібно взаємодіяти один з одним, наприклад, один процес може передавати дані іншому процесу, або кілька процесів можуть обробляти дані із загального файлу. У всіх цих випадках виникає проблема синхронізації процесів, яка може вирішуватися припиненням і активізацією

процесів, організацією черг, блокуванням і звільненням ресурсів.

Отже, операційна система виконує наступні основні функції, пов'язані з керуванням процесами й завданнями:

- створення й видалення завдань;
- планування процесів і диспетчеризація завдань;
- синхронізація завдань, забезпечення їх засобами комунікації.

Створення завдання сполучене з формуванням відповідної інформаційної структури (дескриптора), а її видалення – з розформуванням. Створення й видалення завдань здійснюється відповідно до запитів від користувачів або від самих завдань. Одне завдання може породити інше нове завдання. При цьому між завданнями з'являються «родинні» відносини. завдання, що породжує, називається «батьком», а породжене завдання – «нащадком». Батько може призупинити або вилучити своє дочірнє завдання, тоді як нащадок не може управляти батьком.

Процесор є одним із самих необхідних ресурсів для виконання обчислень. Тому способи розподілу часу центрального процесора між завданнями сильно впливають і на швидкість виконання окремих обчислень, і на загальну ефективність обчислювальної системи. Основним підходом в організації того або іншого методу керування процесами, що забезпечує ефективне завантаження ресурсів або виконання яких-небудь інших цілей, є організація черг процесів і ресурсів. При розподілі процесорного часу між завданнями також використовується механізм черг.

Розв'язок питань, пов'язаних з тим, якому завданню слід надати процесорний час у цей момент, покладається на спеціальний модуль операційної системи, називаний *диспетчером завдань*. Питання ж добору обчислювальних процесів, які не тільки можна, але й доцільно вирішувати паралельно, покладають на *планувальник процесів*.

Найпростішу модель процесу можна побудувати виходячи з того, що в будь-який момент часу процес або виконується, або не виконується, тобто має тільки два стани. Якби всі процеси були б завжди готові до виконання, то черга за цією схемою могла б працювати цілком ефективно. Така черга працює за принципом обробки в порядку вступу, а процесор обслуговує наявності круговим методом (Round-robin). Кожному процесу приділяється певний проміжок часу, після закінчення якого він вертається в чергу.

Однак у такому простому прикладі подібна реалізація не є адекватною: частина процесів готова до виконання, а частина заблокована, наприклад, через очікування вводу-виводу. Тому при наявності однієї черги диспетчер не може просто вибрати для виконання перший процес із черги. Перед цим він повинен буде переглядати весь список, відшуковуючи незаблокований процес, який перебуває в черзі далі від інших. Звідси представляється цілком природно розділити процеси, які не виконуються, на два типа: готові до виконання і заблоковані. Виходячи з цього, корисно додати ще два стани, як показано на

рисунку 2.2.

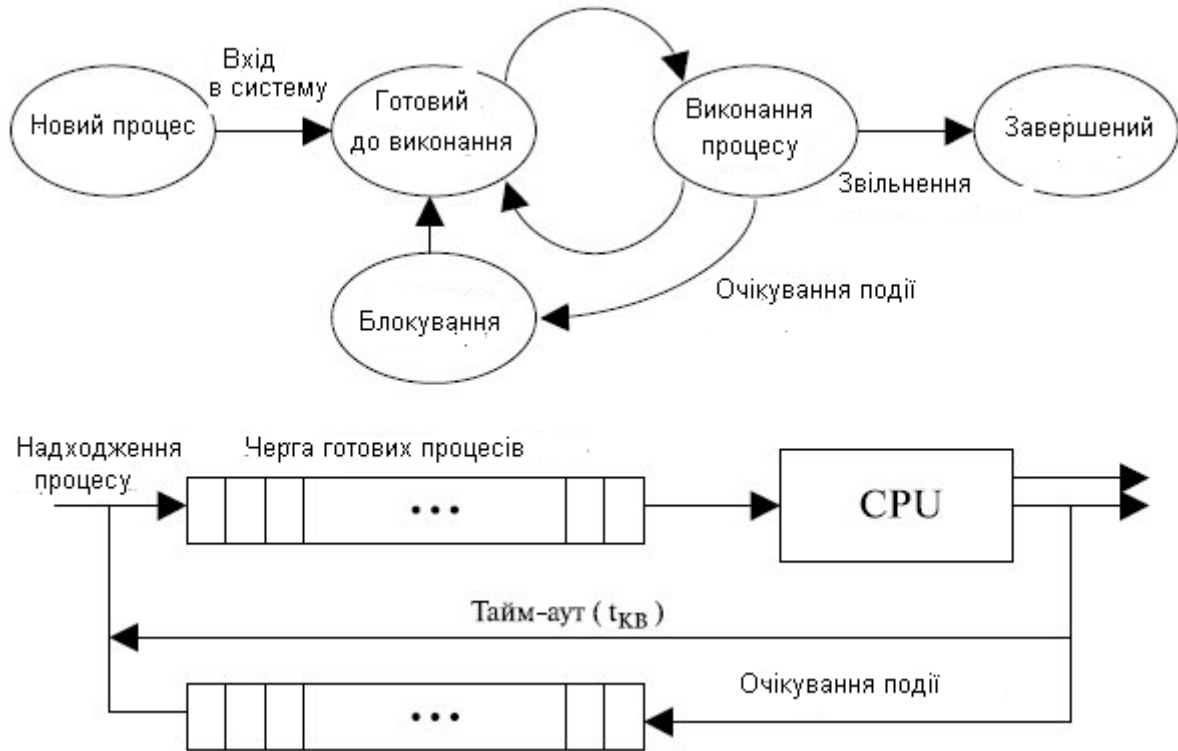


Рисунок 2.2 – Схема виконання процесів

Моделювання процесів

Завдання моделювання процесів слід вирішувати з урахуванням поетапних перетворень даних. Із цією метою необхідно розробити інформаційну модель системи ЧПУ, по якій можна було б оцінювати:

- зміст вхідної інформації;
- зміст вихідної інформації;
- математичний апарат або алгоритм перетворень однієї інформації в іншу (концептуальну модель).

Для побудови інформаційної моделі необхідно провести:

- аналіз інформації, необхідної для керування процесом і для оцінки стану технологічного встаткування;
- аналіз керуючих впливів, які забезпечують необхідну якість технологічного процесу й роботи встаткування;
- аналіз проміжної інформації, оброблюваної в СЧПУ.

Аналіз технологічних процесів і роботи встаткування доцільно проводити в наступній послідовності:

- виділити вузли встаткування й у кожному вузлі визначити виконавчі й інформаційні пристрої;
- установити мову й засіб керування для кожного вузла;

- скласти списки (таблиці) вхідних, вихідних, аварійних і блокувальних сигналів, а також сигналів переривань;
- визначити масиви, у які буде передаватися вихідна інформація.

Після проведення аналізу складається схема зв'язків між усіма компонентами інформаційних масивів. Лінії цих зв'язків позначають конкретні завдання прикладного програмного забезпечення.

2.2 Склад інформаційної моделі

При проектуванні інформаційної моделі важливо чітко представляти: як формується інформація, як вона перетворюється й у якому виді використовується.

Одним із джерел інформації системи ЧПУ є пульт оператора. Він містить клавіатуру й засоби введення управляючої програми.

Образ клавіатури розміщується в пам'яті в табличному виді. Інформація, яка перебуває в такій таблиці, повинна мати адреси пам'яті, за якими містяться інструкції з подальших дій.

Управляюча програма вводиться в пристрій ЧПУ із програмоносія, наприклад, з флешки, або по каналу зв'язку. Введена програма аналізується на предмет наявності помилок і розміщується в пам'яті.

Джерелами вхідної інформації є також датчики стану об'єкта.

Вхідна інформація від пульта оператора й датчиків утворює *перший масив – масив джерел*.

Процес перетворення інформації від пульта оператора й датчиків повинен завершитися підготовкою системи керування до роботи в заданому режимі. Для цього створюється образ режиму. Образ режиму роботи верстата відображається у вигляді *другого масиву – об'єктного*.

При роботі в автоматичному режимі інформація про параметри процесу обробки повинна обновлятися безупинно. Для виключення затримок в управлінні координатними рухами в процесі відпрацьовування поточного кадру повинен бути введений і підготовлений до роботи наступний кадр. У зв'язку із цим треба передбачити *завантажувальний масив* даних.

У процесі програмного керування ведуться інтерполяційні розрахунки й обчислюються завдання на керовані приводи. Інформація цього процесу представляє *головний масив*.

І, нарешті, усі керуючі сигнали на автоматику, електроприводи, засоби візуалізації утворюють *масив споживача*.

Отже, інформаційна модель повинна містити: масив джерел, об'єктний масив, завантажувальний масив, головний масив і масив споживача.

Розділивши масиви на окремі інформаційні компоненти, можливо скласти схему їх зв'язків (рис. 2.3).

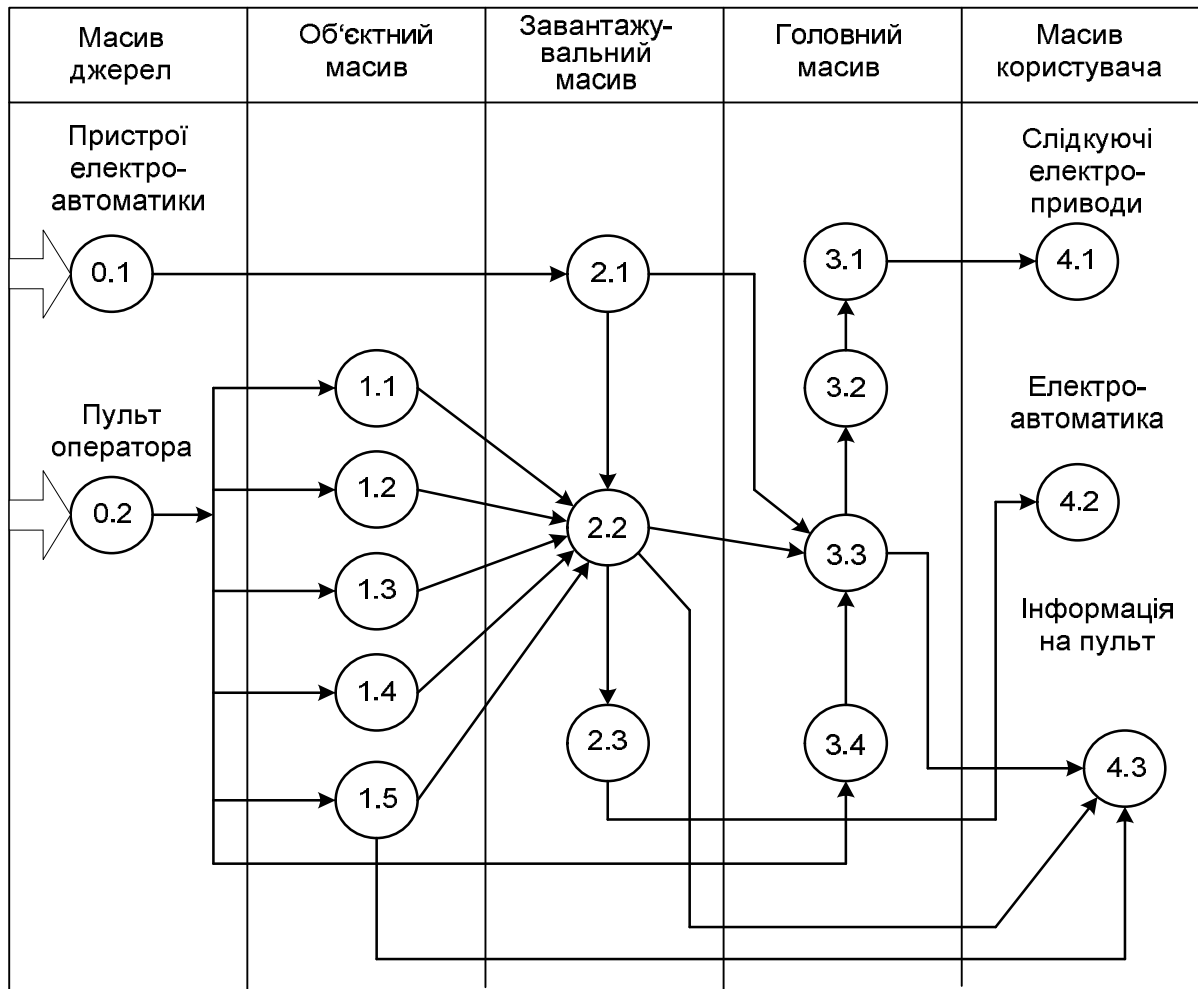


Рисунок 2.3 - Структура інформаційної моделі верстата із ЧПУ

У складеній інформаційній моделі верстата із ЧПУ процеси перетворення одних векторів в інші описуються в такий спосіб.

Вектор вхідної інформації (01) від пристроїв електроавтоматики (подорожніх і кінцевих вимикачів, датчиків) бере участь у формуванні вектора стану системи (2.1), який використовується для формування вектора робочого кадру (3.3) і для обліку блокувань і захистів.

Вектор вхідної інформації (02) від пульта оператора використовується у формуванні вектора режимів (1.1), вектора завдань (1.2), вектора кадру (1.3), вектора корекцій положення інструмента (1.4), вектора візуалізації (1.5), а також вектора корекції режимів (3.4).

Вхідна інформація необхідна також для формування вектора буферного кадру (2.2), векторів керування автоматикою (2.3, 4.2) і засобів індикації (4.3), а також уведення корекції координат у вектор робочого кадру (3.3).

Вектор буферного кадру (2.2) разом з вектором стану автоматики (2.1)

формує вектор інструкцій робочого кадру (3.3), згідно з яким виконуються інтерполяційні розрахунки й задаються швидкості руху (вектор 3.2), а потім формуються керуючі впливи на приводи (3.1, 4.1).

Перетворення масивів здійснюється двома шляхами:

- асинхронним автоматом, коли перехід до нового масиву здійснюється по досягненню необхідних умов;
- системою таймерних переривань, коли керування процесами перетворення інформації здійснюється в режимі реального часу.

У системі переривань вищі рівні пріоритетів віддаються завданням, які вимагають швидкої реакції – це обслуговування аварійних сигналів, керування автоматикою й приводами. Далі пріоритети розподіляються на завдання обслуговування панелі оператора, формування буферного кадру, обміну даними з верхнім рівнем і т.п.

Після створення інформаційної моделі розробляються концептуальні (математичні) моделі процесів перетворення інформації одного вектора в інформацію другого вектора.

2.3 Система ЧПУ й об'єкт управління як функціональний автомат

Функціональний автомат (ФА) являє собою сукупність управляючого автомата (УА) і операційного автомата (ОА). Структурна схема функціонального автомата представлена на рисунку 2.4.

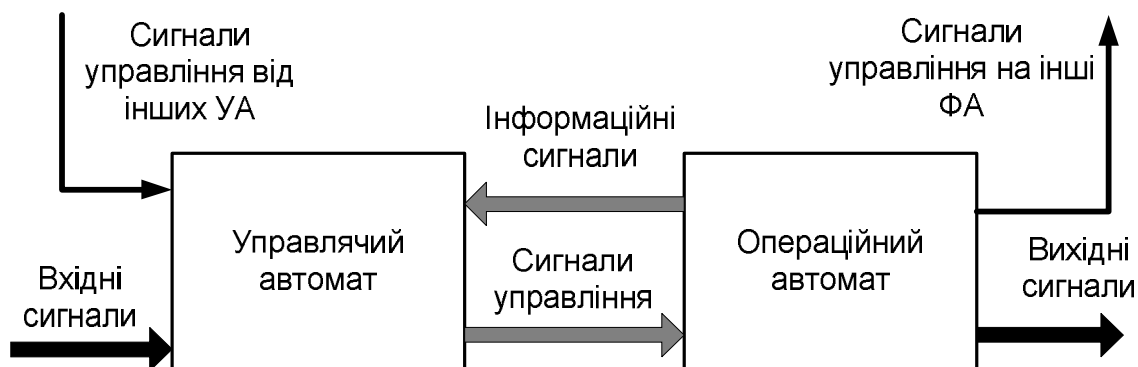


Рисунок 2.4 - Структурна схема функціонального автомата

Вхідні сигнали керування ФА являють собою *вхідний код*, вихідні сигнали ФА – *вихідний код*. Функціональний автомат, таким чином, здійснює перетворення одного коду в інший код, тобто трансляцію.

Трансляція може бути організована двома типами процесів:

1. Компіляцією, коли вся інформація вводиться, осмислюється й перетворюється в іншу мову (як переклад тексту).
2. Інтерпретацією, коли переклад на іншу мову здійснюється порціями (фразами) у міру вступу інформації.

У пристроях ЧПУ при введенні управляючої програми застосовується *інтерпретація*.

Структурна схема інтерпретатора показана на рисунку 2.5.

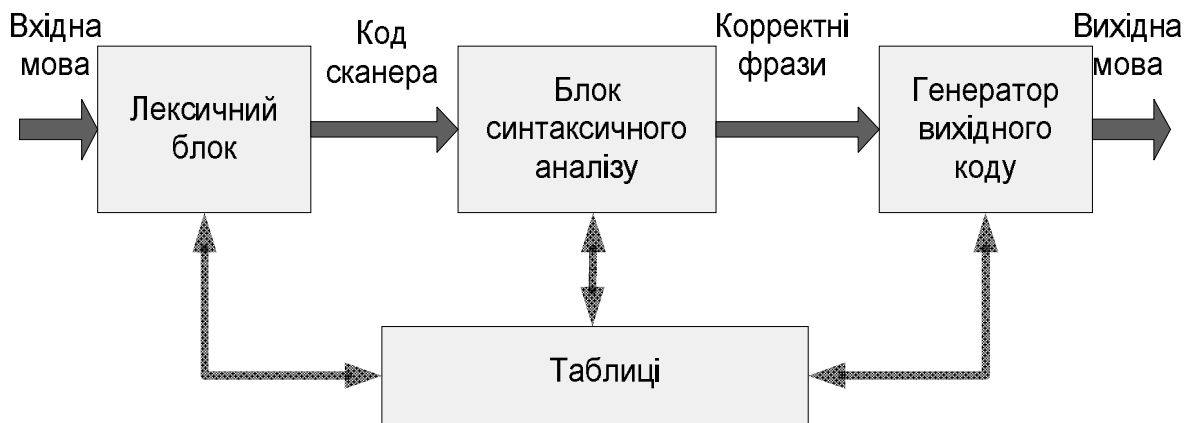


Рисунок 2.5 - Структурна схема інтерпретатора

Робота інтерпретатора заключається в наступному. Управляюча програма, яка написана на деякій вхідній мові, сканується з метою визначення приналежності терміналів цієї мови (лексичних змінних, або лексем) прийнятому в даній системі формату програми.

Усі припустимі в уведеній програмі термінали (символи) впаковуються в спеціальні таблиці й забезпечуються відповідною адресою – кодом сканера.

Код сканера дозволяє відновити або розподілити введenu інформацію при проведенні синтаксичного аналізу, коли визначається правильність формату кадру, а також при проведенні семантичного аналізу, коли визначається зміст завдання. Сформовані при синтаксичному аналізі коректні фрази надходять на генератор вихідного коду, який подає цей код на автомат семантичного аналізу.

2.4 Синтаксичний аналіз управляючої програми системи ЧПУ

Правила написання слів і речень визначаються граматиною вхідної мови. У системах ЧПУ застосовуються автоматна й контекстно-вільна грамика. Автоматна грамика передбачає жорсткі правила написання тексту програми. У контекстно-вільній грамиці припустима деяка свобода в послідовності слів або їх розмірності. Наприклад, функція подачі F може не містити

незначущих цифр і кома може стояти на будь-якій місці, наприклад, F0,25; F1250; F1,2 і т.д.

Окремі фрази вхідної мови являють собою команди управління. Однак їх набір може бути суперечливим, невірним або неповним. Із цією метою проводиться синтаксичний і семантичний аналіз.

Для синтаксичного аналізу правила граматики зручно представляти у вигляді синтаксичної діаграми, приклад якої наведено на рисунку 2.6.

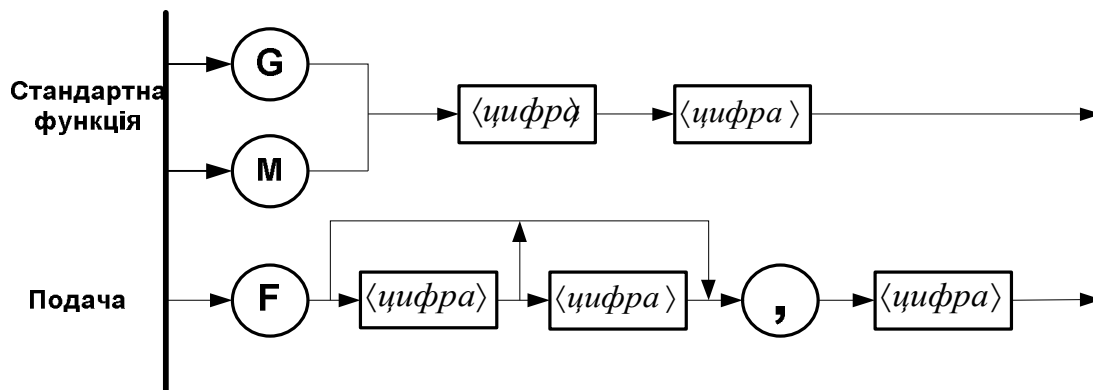


Рисунок 2.6 - Приклад синтаксичної діаграми

Синтаксичні діаграми мають вхід (ліва вертикаль) і вихід (права вертикаль). Між ними розташовуються припустимі ланцюжки *терміналів* (у кружках) і станів (у прямокутниках). Синтаксичні діаграми показують варіанти вистави функцій. Наприклад, у функції «Подача» повинні бути термінал F, термінал кома (,) і одна, дві або три цифри. Для виходу із процесу аналізу необхідна також команда введення наступного символу – SIMBLE.

Переходи в нові стани повинні відрізнятися терміналами. Так, наприклад, після введення терміналу F є два варіанта наступного кроку: введення цифри або введення коми.

Представимо синтаксис слова «Подача» графом, який показаний на рисунку 2.7. Початковий стан на цьому графі позначений символом « σ ». Переходи із одного стану в інший здійснюються при наявності на вході певного терміналу.

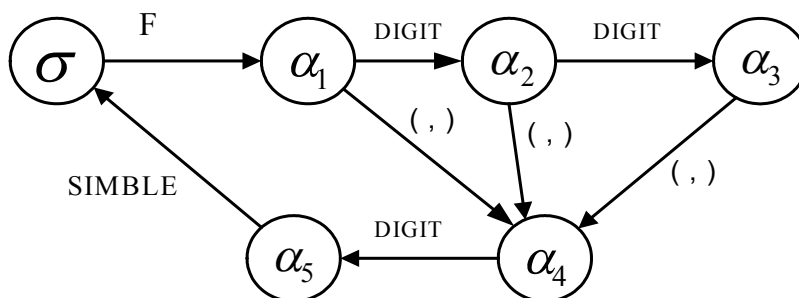


Рисунок 2.7 - Граф автомата для формування функції «Подача»

Від графа перейдемо до складання правил *автоматної* граматики, які записуються у вигляді:

$$\alpha \rightarrow \eta \mu \xi,$$

де α – корінь правила, нетермінал, який позначає поточний стан автомата; η – термінал, один, як умова для переходу в новий стан; μ – нетермінал, стан, у який переходить автомат при виконанні умови η ; ξ – сигнал $\{z_i\}$, який генерується автоматом у новому стані (не обов'язковий).

Для графа на рисунку 2.7 правила автоматної граматики ухвалюють наступний вид:

$$P = \left\{ \begin{array}{l} \sigma \xrightarrow{1} F\alpha_1; \alpha_1 \xrightarrow{2} DIGIT\alpha_2; \alpha_1 \xrightarrow{3} (,)\alpha_4; \alpha_2 \xrightarrow{4} DIGIT\alpha_3; \\ \alpha_2 \xrightarrow{5} (,)\alpha_4; \alpha_3 \xrightarrow{6} (,)\alpha_4; \alpha_4 \xrightarrow{7} \alpha_5; \alpha_5 \xrightarrow{8} SIMBLE\sigma \end{array} \right\}$$

Складені правила дозволяють розробити програму автомата.

Сигналу z можна призначити різні значення, наприклад, $z=1$ – ДОПУСТИТИ; $z=0$ – ВІДКИНУТИ.

Алгоритм управління процесом синтаксичного аналізу фрази «Подача» наведений у вигляді таблиці (табл. 2.1).

Таблиця 2.1 - Алгоритм роботи автомата при аналізі слова «Подача»

Стан	Термінал			
	F	DIG	< , >	Команда SIMBLE
σ	Замінити α_1 зрушити			
α_1		Замінити α_2 зрушити	Замінити α_4 зрушити	
α_2		Замінити α_3 зрушити	Замінити α_4 зрушити	
α_3			Замінити α_4 зрушити	
α_4		Замінити α_5 зрушити,		
α_5				Замінити σ зрушити

У керуючій таблиці автомата рядки описують варіанти його обігу в певних станах, а стовпці – умови переходів з одного стану в інший.

Правильність складання таблиці оцінюється в такий спосіб:

- 1) кількість правил рівняється кількості заповнених кліток таблиці;
- 2) незаповнені клітки відповідають операції «ВІДКИНУТИ»;
- 3) кількість рядків відповідає числу станів, а кількість стовпців – числу терміналів.

2.5 Аналіз кадру управляючої програми

Аналіз кадру, написаного в контекстно-вільній граматиці, здійснюється за допомогою автоматів з магазинною (стековою) пам'яттю – МП-автоматів.

На відміну від автоматної граматики в правій частині правил контекстно-вільної граматики допускається ланцюжок терміналів, а також ланцюжок нетерміналів, можливо порожній, наприклад:

$$\alpha_1 \rightarrow T_1 T_2 \alpha_2; \quad \alpha_2 \rightarrow T_6; \quad \alpha_1 \rightarrow T_1 \alpha_3 T_5 \alpha_6.$$

У складанні правил контекстно-вільної граматики є два обмеження:

- права частина правила повинна починатися з *термінала*;
- для однакових нетерміналів у лівій частині різні правила повинні мати різні перші термінали в правій частині, наприклад:

$$\alpha_1 \rightarrow T_1 \alpha_3 T_5 \alpha_6, \quad \alpha_1 \rightarrow T_4 \alpha_5.$$

МП-автомат задається трьома множинами, які розміщуються в пам'яті.

1. Множина вхідних символів T_i (терміналів), а також кінцевий маркер « \perp » (повернута на 90° буква T), що вказує на закінчення ланцюжка терміналів, які надходять зі сканера.

2. Множина магазинних символів, в яку входять:

- нетермінали, які завантажуються в магазин для запам'ятовування «маршруту» аналізу й потім виштовхуються з магазину за правилами граматики;
- термінали із правил, які записані в правилах не на першому місці;
- маркер дна магазину ∇ .

3. Множина керуючих впливів, які повинен згенерувати МП-автомат.

При роботі МП-автомата виконуються стандартні процедури:

1) операції, виконувани над входом:

ЗРУШИТИ, тобто перейти до аналізу наступного термінала;

ЗАТРИМАТИ, тобто поточний термінал затримується для участі в на-

ступному циклі роботи;

2) операції над магазинною пам'яттю:

ВИДАЛИТИ, тобто зі стекової пам'яті (магазину) видаляється верхній символ – термінал або нетермінал;

ЗАМІНИТИ – з магазину видаляється верхній символ і замість нього вводиться інший символ або ланцюжок символів;

3) службові операції:

ВІДКИНУТИ – рівносильна сигналу «Помилка введення» (усі порожні місця в керуючій таблиці відповідають цій операції).

ДОПУСТИТИ – МП-автомат працює як розпізнавач у дотриманні правил граматики і видає такий сигнал при успішному результаті.

ВИДАТИ $\{zi\}$ – автомат генерує сигнал управління, якщо ланцюжок вхідних терміналів закінчився кінцевим маркером, а верхній символ стекової пам'яті є маркер дна магазину.

Робота автомата закінчується тоді, коли проаналізовані всі термінали, які поступили на вхід, і маркер дна магазину перебуває вгорі.

2.6 Приклад проектування управляючої таблиці МП-автомата

Нехай правила граматики мають вигляд:

$$P = \{ \sigma \xrightarrow{1} T_1 T_2 \alpha_1; \sigma \xrightarrow{2} T_2 \alpha_1 T_2 \sigma; \alpha_1 \xrightarrow{3} T_3 \alpha_2; \alpha_2 \xrightarrow{4} T_4 T_5 \alpha_1; \alpha_1 \xrightarrow{5} T_6 \}$$

Складання управляючої таблиці ведеться в наступній послідовності:

- правила представляються у формі $\alpha = \eta \lambda$, де η – перший термінал; λ – інша частина терміналу;
- складається підготовча таблиця (табл. 2.2);

Таблиця 2.2 - Підготовка даних для керуючої таблиці

№ правила	Корінь правила	Перший термінал η	Інша частина правила λ
1	σ	T_1	$T_2 \alpha_1$
2	σ	T_2	$\alpha_1 T_2 \sigma$
3	α_1	T_3	α_2
4	α_2	T_4	$T_5 \alpha_1$
5	α_1	T_6	-

- складається список вхідних символів: $T_1; T_2; T_3; T_4; T_5; T_6; \neg$;
- складається список магазинних символів: $\sigma; \alpha_1; \alpha_2; T_2; T_5; \nabla$;
- будується управляюча таблиця (табл. 2.3).

Таблиця 2.3

Магаз. символи	Вхідні символи						
	T1	T2	T3	T4	T5	T6	\neg
σ	Замінити на ($\alpha_1 T_2$) зрушити	Замінити на ($\sigma T_2 \alpha_1$) зрушити					
α_1			Замінити на (α_2) зрушити			Видалити зрушити	
α_2				Замінити на ($\alpha_1 T_5$) зрушити			
T_2		Видалити зрушити					
T_5					Видалити зрушити		
∇							Допустити

При побудові управляючої таблиці МП-автомата необхідно враховувати наступні правила:

1. Процедура «ВІДКИНУТИ» не записується, їй відповідають порожні клітки.
2. На перетинанні рядків і стовпця, які відповідають певному правилу, частина правила λ записується реверсивно, тобто у зворотному порядку:

$$T_2 \alpha_1 \rightarrow \alpha_1 T_2 .$$

3. Якщо верхній магазинний символ (термінал) збігається із вхідним символом, то виконуються операції ВИДАЛИТИ й ЗРУШИТИ.

4. Якщо верхній символ магазину ∇ й вхідний символ \neg (маркери), виконується операція ДОПУСТИТИ.

Приклад.

Нехай на вхід МП-автомата надходить ланцюжок терміналів: $T_2 T_3 T_4 T_5 T_6 T_2 \neg$.

Розглянемо процес зміни інформації в магазині. Процедури, виконувані автоматом при аналізі кожного терміналу, представлено в таблиці 2.4.

Таблиця 2.4 - Приклад аналізу вхідної інформації

Магазин низ верх	Вхід	Процедура
$\nabla\sigma$	T_2	Замінити верхній символ на $T_2\alpha_1$, зрушити
$\nabla T_2\alpha_1$	T_3	Замінити на α_2 , зрушити
$\nabla T_2\alpha_2$	T_4	Замінити на α_1T_5 , зрушити
$\nabla T_2\alpha_1T_5$	T_5	Видалити, зрушити
$\nabla T_2\alpha_1$	T_6	Видалити, зрушити
∇T_2	T_2	Видалити, зрушити
∇	— 	Допустити

Аналіз вхідного ланцюжка терміналів МП-автомата показує, що даний ланцюжок являє собою коректну фразу, яка відповідає правилам.

Таким чином, процес трансляції полягає в читанні інформації за допомогою правил граматики й формуванні керуючих сигналів у операційному автоматі – генераторі кодів. Генератор кодів програмно реалізує завдання логічного керування автоматикою.

3 МЕТОДИ АЛГОРИТМІЗАЦІЇ ЗАВДАНЬ УПРАВЛІННЯ АВТОМАТИКОЮ

Управління автоматикою у верстатах із ЧПУ здійснюється контролерами автоматики. Програмне забезпечення контролерів автоматики розробляється з використанням мов програмування, рекомендованих міжнародною електричною комісією (МЕК), яка розробила стандарт ІЕС 6-1131/3. Ці мови програмування поєднують у собі функціональність і простоту. Вони також охороняють користувача від більшості помилок, які нерідко виникають при використанні звичайних мов програмування.

Для всіх шести мов існує єдиний механізм *зв'язку з базою даних реального часу*. Відповідно до цього механізму кожна програма повинна мати набір аргументів. Початкові дані передаються в програму через вхідні аргументи, а результати обчислень вертаються у вихідних аргументах. Аргументи зв'язуються з атрибутами каналів, тобто з реальними входами й виходами контролерів, із пристроями введення-виводу, осередками корпоративних баз даних, а також із внутрішніми змінними. Завдяки такій схемі та сама програма може викликатися кілька раз за цикл для обробки різних потоків даних.

Програмування й налагодження програм на мовах ІЕС 6-1131/3 проводиться в *інтегрованій середовищі розробки*, що включає в себе декілька різних редакторів. Програми на мовах FBD, LD, SFC і CFC створюються й налагоджують у спеціальних *візуальних редакторах*, а ST і IL – у *текстових редакторах*.

В цих мовах використовуються відомі широкому колу спеціалістів по автоматизації процесів графічні нотації у вигляді електричних схем мікроелектроніки (FBD), схем релейної автоматики (LD), блок-схем алгоритмів та мережі Петрі (SFC) а також структурні схеми (CFC).

При керуванні простими процесами, а також при аналізі станів з невеликою кількістю змінних (3...8) функції керування можуть мати різні способи вистави: булеві вираження, таблиці істинності, перемикальні або комбінаційні логічні схеми, карти Карно, діаграми Вейча, графи і т.д. Вихідні сигнали управління можуть бути однобітними або багатобітними. Для аналізу станів і формування вихідних сигналів застосовуються методи операторних програм, у яких використовуються логічні операції процесорів, а також методи маскування, адресних переходів і бінарних програм.

3.1 Застосування методу маскування

Якщо розв'язок функції повинен бути один із двох варіантів $F \in [0;1]$, то при її завданні в булевій формі або у вигляді перемикальної схеми можна

застосувати метод відображення вхідного набору або метод маскування.

Нехай булева функція має вид: $F = x_1 x_3 \bar{x}_5 \bar{x}_6 \bar{x}_7 \vee \bar{x}_2 x_3 x_4$.

Представимо функцію у вигляді двох функцій: $F = Y_1(x) \vee Y_2(x)$,

де $Y_1(x) = (-000-1-1)$; $Y_2(x) = (----110-)$.

У наведених вираженнях риси вказують на відсутність умов, а 0 і 1 значають стани змінних. При цьому змінні розміщені в байті в такий спосіб: $[x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1]$.

Помістимо $Y_1(x)$ і $Y_2(x)$ у сусідні комірки пам'яті й підготуємо маски для аналізу інформації.

Перша маска G повинна закрити всі змінні, які не несуть умов:

- для $Y_1(x)$ $G_1 = (01110101) = 75H$;
- для $Y_2(x)$ $G_2 = (00001110) = 0CH$.

Друга маска Γ (*гама*) повинна виділити одиничні значення в Y :

- для $Y_1(x)$ $\Gamma_1 = (00000101) = 05H$;
- для $Y_2(x)$ $\Gamma_2 = (00001100) = 0CH$.

Маски розміщуються в пам'яті попарно і послідовно: $G_1, \Gamma_1, G_2, \Gamma_2$.

Програмна реалізація автомата полягає в тому, що на вхідний набір накладається маска G_1 за допомогою логічної операції «І» – $ANI G_1$, а потім маска Γ_1 операцією «Виключаюче АБО» – $XRI \Gamma_1$. Якщо вхідний набір задовольняє умовам $Y_1(x)$, то в регістрі станів процесора встановлюється біт нульового результату і командою умовного переходу здійснюється вивід значення функції $F=1$. Якщо умови не задоволені (біт нульового результату рівний 0) при перевірці обох функцій, то виводиться результат $F=0$.

Функція може містити кілька операцій АБО. Нехай, наприклад, функція містить дві операції АБО:

$$F = x_1 x_2 x_4 \bar{x}_6 \vee x_3 \bar{x}_4 x_7 \vee x_5 x_8$$

У цьому випадку буде потрібно представити її у вигляді $F = Y_1(x) \vee Y_2(x) \vee Y_3(x)$ й одержати три пари масок.

Гідність методу – програма універсальна, тому що при зміні функції досить змінити маски.

Недолік методу – при великому числі змінних і кон'юнкцій значно збільшується кількість масок.

3.2 Метод маскуванню багатомісних логічних функцій

У практиці програмування часто зустрічаються ситуації із труднощами завдання умов і вибору розв'язків. Поведінка автомата описується при цьому матрицею багатомісних логічних функцій, у якій умови переходу задаються матрицями умов $U_{i,j} \in \{1,0,-\}$, а результати роботи представляються матрицею розв'язків $D_{i,j} \in \{0,1,-\}$, де прочерки позначають відсутність умов або розв'язків.

При представленні процесу графом перехід з одного стану (S_1) в інший (S_2) здійснюється по дузі x_i (рис. 3.1).

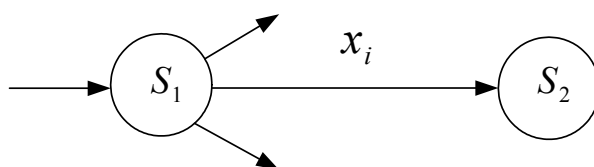


Рисунок 3.1 - Позначення переходу автомата в новий стан

Для реалізації процесу необхідно знати для кожного стану:

- вектор умов переходів у інші стани $[x_i]$;
- вектор розв'язків $[y_i]$ – інформацію про те, які сигнали автомат повинен установити, які зняти і які залишити без змін у новій вершині;
- адресу пам'яті, де перебувають дані про вектори умов і вектори розв'язків для нової вершини.

Таким чином, програмна реалізація автомата вимагає організації двох циклів. Оскільки з вершини (стану) може виходити кілька дуг, то в **першому циклі** здійснюється пошук дуги, яка задовольняє умовам, що втримуються у вхідному наборі. Якщо жодна умова не підходить, автомат може вийти в непередбачений стан. Щоб виключити цей варіант уводиться фіктивна дуга з незначимими параметрами (— — — — —), яка замикає перехід на цю саму вершину, зберігаючи тим самим стан автомату.

У **другому циклі** здійснюється аналіз вектора розв'язків і формується вихідний сигнал автомата згідно вектору розв'язків. Після цього здійснюється читання інформації нової адреси в області пам'яті, де зберігаються умови й інформація для наступних циклів аналізу.

Аналіз умов і розв'язків здійснюється за допомогою маскуванню. Для кожної дуги потрібно створити чотири маски. Позначимо маски для вектора умов $M^1_{K(x)}$ і $M^2_{K(x)}$, а для вектора розв'язків, відповідно, $M^1_{K(y)}$ і $M^2_{K(y)}$, де

$k=0, 1, \dots, 7$ – номер розряду маски.

Перша маска виділяє значимі розряди байту вектора умов:

$$M_{K(x)}^1 = \begin{cases} 1, & \text{при } r_k = (0 \vee 1) ; \\ 0, & \text{при } r_k = (-). \end{cases}$$

Друга маска виділяє одиничні значення:

$$M_{K(x)}^2 = \begin{cases} 0, & \text{при } r_k = (- \vee 0) ; \\ 1, & \text{при } r_k = 1. \end{cases}$$

Третя маска виділяє розряди, у яких вихідні сигнали не повинні змінюватися:

$$M_{K(y)}^1 = \begin{cases} 0, & \text{при } r_k = (1 \vee 0) ; \\ 1, & \text{при } r_k = -. \end{cases}$$

Четверта маска виділяє розряди, які необхідно встановити в одиницю.

Інші розряди скидаються:

$$M_{K(y)}^2 = \begin{cases} 0, & \text{при } r_k = (0 \vee -) ; \\ 1, & \text{при } r_k = 1. \end{cases}$$

Операції з масками аналогічні тим, які були наведені вище.

Нехай дуга 01 представлена вектором умов (0–1010–1), вектором розв'язків (01—001–) і адресою 0926H.

При програмуванні автомата інформація для цієї дуги буде розміщена в пам'яті в наступній послідовності:

ДУГА 01: 10111101 $\rightarrow M_{K(x)}^1=BDH$;
00101001 $\rightarrow M_{K(x)}^2=29H$;
00110001 $\rightarrow M_{K(y)}^1=31H$;
01000010 $\rightarrow M_{K(y)}^2=42H$;
00100110 (молодший байт адреси 26H);
00001001 (старший байт адреси 09H).

Таким чином, для опису однієї дуги потрібно шість байт пам'яті.

3.3 Формалізм опису складних автоматичних циклів

Формалізм ієрархічних графів

Для опису електроавтоматики можна використовувати формалізм ієрархічних графів, який особливо зручний для *графічного* опису *автоматичних циклів*. Ієрархічний граф являє собою «четвірку» множин:

- *Прості вершини-стану*, зображувані *кружками*, причому стани можуть бути статичними або динамічними. Вихід зі статичного стану ініціюється ззовні, тоді як вихід з динамічного стану відбувається по завершенню процесу.

- *Складні вершини-стану*, зображувані *подвійними кружками* (з подвійним бордюром), причому такі стани є *вкладеними* графами.

- *Дуги*, які відбивають переходи між станами будь-якого типу.

- *Вузли*, які розрізають дуги, *зображуються темними кружками*, причому вузли фіксують умови зміни станів будь-якого типу. Якщо дуга виходить зі статичної вершини-стану, то вузол може одночасно належати іншому графові того ж або іншого рівня ієрархії.

Вершинам-станам приписані «етикетки» – імена в прямокутних рамках.

Ім'я статичного стану має структуру СТАТУС_<ім'я>.

Ім'я динамічного стану (процесу або ініціалізації) має структуру ПРОЦ_<ім'я> або ІНЦ_<ім'я>.

Ім'я вузла служить ознакою одного з наступних типів:

- команди панелі оператора;
- прапорця завершення обчислювального процесу;
- сигналу закінчення керованої операції.

Методика опису складного циклу електроавтоматики включає в собі наступні етапи:

- розробка первинного автомата, тобто автомата верхнього рівня ієрархії, який є, по суті, диспетчером режимів;

- розробка режиму нерегулярних ситуацій (внутрішнього режиму), який зберігає коректність стану керованого об'єкта при будь-яких перемиканнях основних режимів, а також гарантує незмінний стан об'єкта, якщо цикл пасивний;

- виділення паралельно працюючих автоматів, які діють у рамках циклу;

- розробка автоматів нижнього рівня ієрархії.

Розглянемо систему керування револьверною головкою токарського верстата (рис. 3.2), яка підтримує наступні режими:

- автоматичний, у рамках якого забезпечується виклик інструмента будь-якої грані револьверної головки (перехід на потрібну грань відбувається по найкоротшому шляху);

- ручного керування (наприклад, поворот револьверної головки на чергову її грань);

- нерегулярних ситуацій.

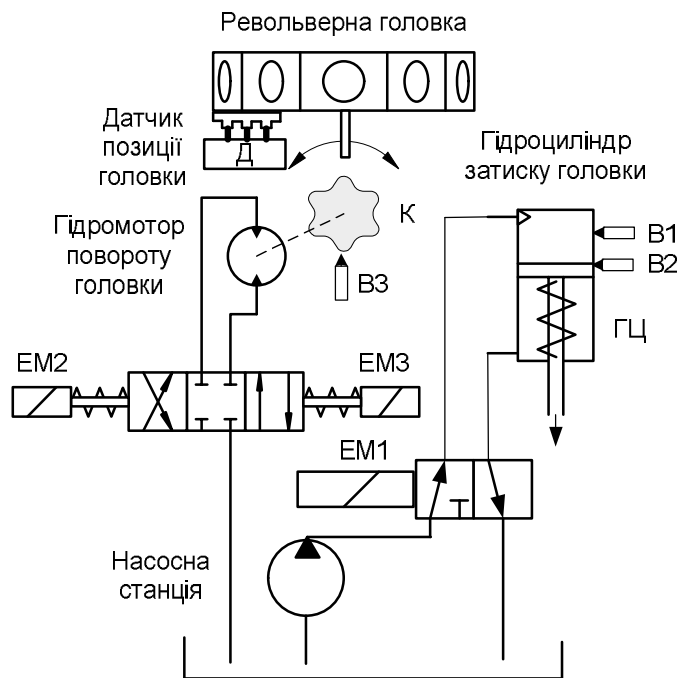


Рисунок 3.2 - Схема засобів керування револьверною головою

Конструкція револьверної головки

Двопозиційний гідророзподільник з електромагнітом ЕМ1 управляє гідроциліндром ГЦ затискача-розтискача револьверної головки, нижнє й верхнє положення якого контролюються вимикачами В1, В2. Поворот головки здійснюється гідромотором ГМ, який управляється гідророзподільником з електромагнітами ЕМ2 і ЕМ3. Проходження будь-якої грані через можливе (по куту повороту) положення ідентифікується кулачком К, співвісним з валом гідромотора, і вимикачем В3. Код грані встановлюється багаторозрядним датчиком позиції головки Д.

На рисунку 3.3 наведений граф **первинного автомата** з усіма складними станами – СТАТУС_НЕРЕГ_РЕЖ (режим нерегулярних ситуацій), СТАТУС_АВТОМ_РЕЖ (автоматичний режим), СТАТУС_РУЧН_РЕЖ (режим ручного керування).

Первинний автомат ініціюється вузлом-умовою (клавiшею) початкового пуску НАЧ_ПУСК, а після цього управляє режимними переходами. Вузли-умови АВТОМАТ і РУЧН відповідають активізації відповідних режимних керуючих елементів панелі оператора.

Вхід у складну вершину СТАТУС_НЕРЕГ_РЕЖ первинного автомата здійснюється через один з вузлів-умов: НАЧ_ПУСК, АВТОМАТ, РУЧН. У цій вершині ідентифікується поточний стан револьверної головки й визначається готовність автомата працювати в ручному або автоматичному режимах.

В автоматичному режимі можна виділити два паралельні процеси – затискання-розтискання головки й обертання-останов головки. Принципом виділення процесу є наявність виконавчого пристрою.

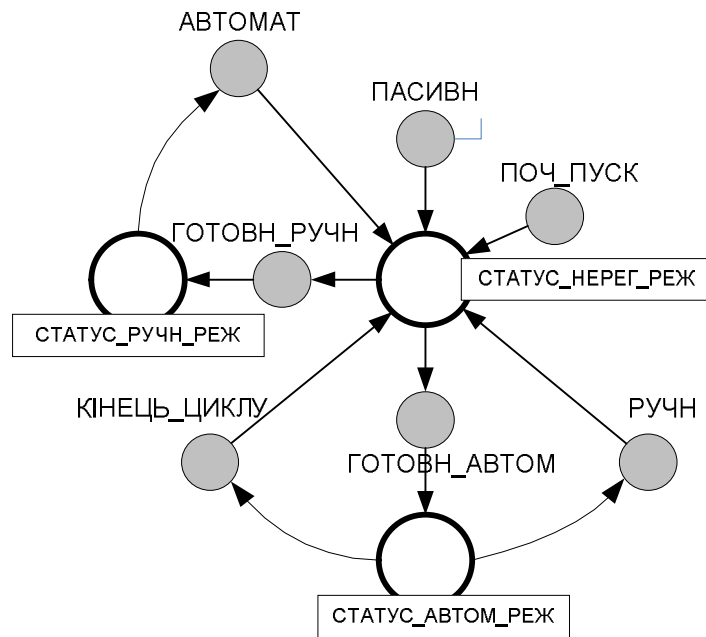


Рисунок 3.3 - Граф первинного автомата

Ініціалізація затискання-розтискання здійснюється в стані ІНІЦ_1 графа (рис. 3.4).

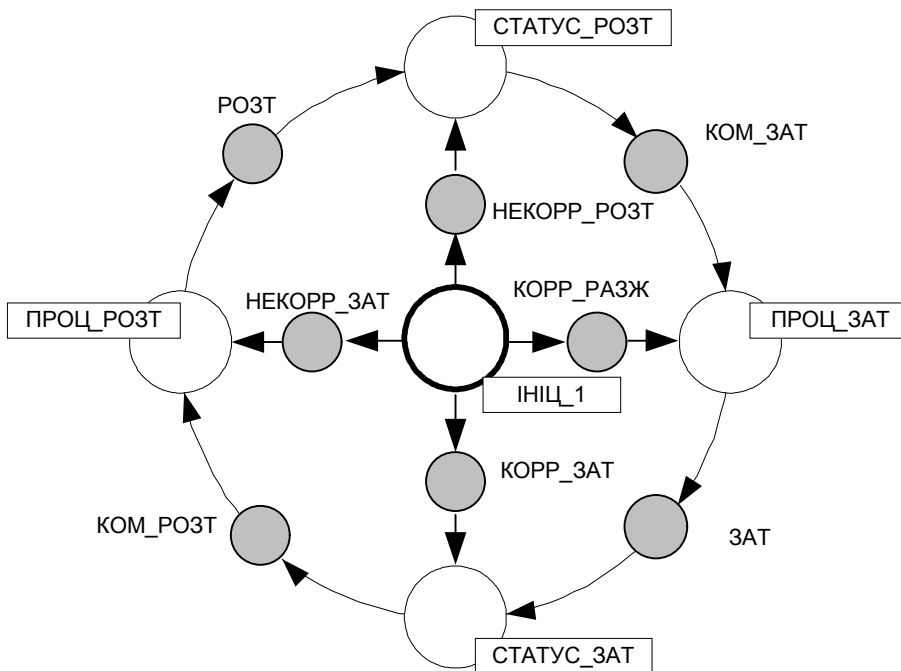


Рисунок 3.4 - Граф автомата для процесу затискання-розтискання

Ініціалізація встановлює умови:

- коректне розтискання (КОРР_РОЗТ), що означає розціплений стан головки в коректнім положенні її граней по куту повороту (умова ініціює процес затискання в стані ПРОЦ_ЗАТ);
- коректне затискання (КОРР_ЗАТ), що означає затиснутий стан го-

ловки в коректнім положенні її граней по куту повороту (умова ініціює перехід у статичний стан процесу затискання СТАТУС_ЗАТ);

- некоректне затискання (НЕКОРР_ЗАТ), що означає затиснутий стан головки в неправильнім положенні її граней по куту повороту (ініціює перехід у стан розтискання головки ПРОЦ_РОЗТ);
- некоректне розтискання (НЕКОРР_РОЗТ), що означає розціплений стан головки в неправильнім положенні її граней по куту повороту (ініціює перехід у статичний стан процесу розтискання СТАТУС_РОЗТ).

Ініціалізація обертання-останова здійснюється в стані ІНІЦ_2 графа (рис. 3.5).

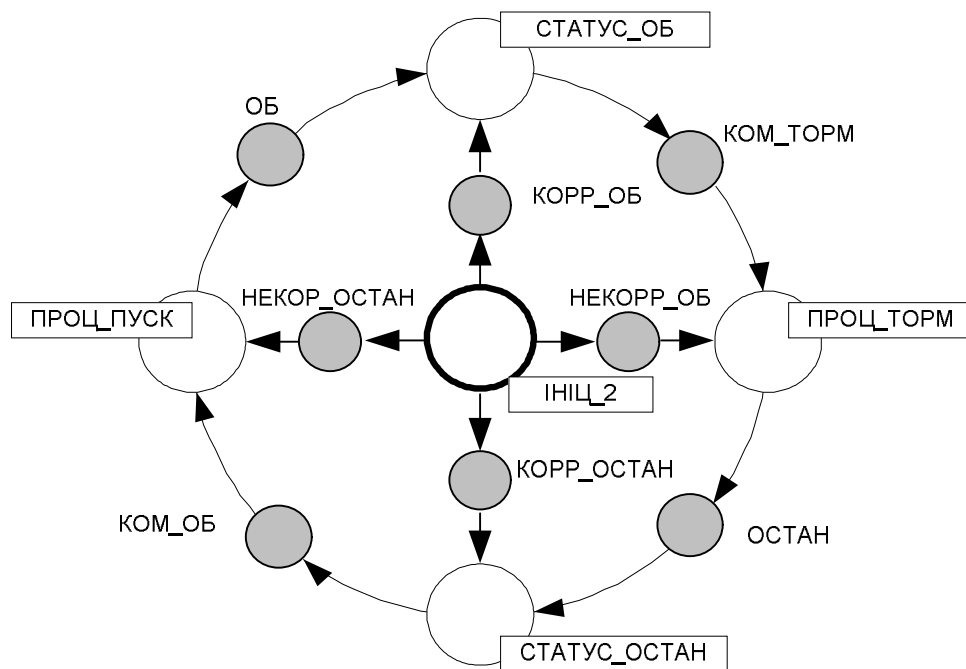


Рисунок 3.5 - Граф автомата для процесу обертання-останова

Ініціалізація встановлює умови:

- некоректне обертання (НЕКОРР_ОБ), що означає включений стан обертання гідромотора ГМ повороту головки при її затиснутому стані через збої (ініціює процес гальмування головки в стані ПРОЦ_ТОРМ);
- коректний останов (КОМП_ОСТАН), що означає зупинку головки в коректнім положенні її граней по куту повороту (визначає перехід у статичний стан СТАТУС_ОСТАН);
- некоректний останов (НЕКОР_ОСТАН), що означає зупинку головки в неправильнім положенні її граней по куту повороту (ініціює процес пуску обертання ПРОЦ_ПУСК);

- коректне обертання (КОРР_ОБ), що означає нормальне обертання головки в її розціпленому стані (визначає перехід у статичний стан СТАТУС_ОБ).

Загальний розгляд двох графів на рисунках 3.3 і 3.4 показує, що система забезпечує автоматичний вихід з будь-яких некоректних ситуацій.

Таким чином, управляючи умовами виходу зі статичних станів, можна побудувати будь-який цикл автоматичного керування.

Для автоматичних циклів доцільно використовувати технології візуального проектування, які дозволяють генерувати файли, що виконуються, по їхній графічній виставі, наприклад, в S7 Hi-Graph.

Функції візуального проектування забезпечують:

- багаторівневе вкладення графів з роботою на кожному рівні в окремій вікні;
- виконання групових операцій (виділення фрагмента графа, видалення, копіювання, переміщення фрагментів у різних позиціях і на різних рівнях);
- збереження (завантаження) проекту або фрагмента;
- імпорт одного проекту в іншій;
- документування проекту й генерацію звітів;
- генерацію вихідного коду для наступної компіляції;
- верифікацію графа на рівні проектування, моделювання й налагодження циклів.

Застосування інструментів візуального проектування багаторазово підвищує продуктивність розроблювача, дозволяє створювати складні цикли електроавтоматики, реалізація яких без інструментальної підтримки проблематична.

3.4 Графічне представлення паралельних процесів мережею Петрі

Керування автоматикою встаткування часто вимагає узгодження керуючих впливів одних систем з іншими, забезпечуючи тим самим певну сукупність паралельних процесів керування.

Алгоритмічний опис паралельних процесів блок-схемами алгоритмів і графами автоматів не дозволяє представляти поділ і об'єднання паралельних процедур. У зв'язку із цим був розроблений інший спосіб графічної вистави алгоритму паралельних процесів у вигляді мережі Петрі.

Мережа Петрі – це математична модель паралельних процесів, взаємодіючих асинхронно.

Теорія мереж Петрі розроблялася рядом авторів з 1962 р., починаючи з Карла Адама Петрі, німецького дослідника. У результаті з'явився клас мереж

Петрі й ряд визначень, як, наприклад, кішка стосовно котячих: автоматна мережа, маркірована, позначена, проста, двоїста і т.д.

З погляду теорії графів мережа Петрі – це орієнтований граф із двома типами вершин, що з'єднуються дугами.

Перший тип вершин називається місцем або позицією й позначається кружком, аналогічно стану автомата на графові, другий тип вершин називається переходом і позначається рисою, яка за змістом позначає бар'єр (рис. 3.6).

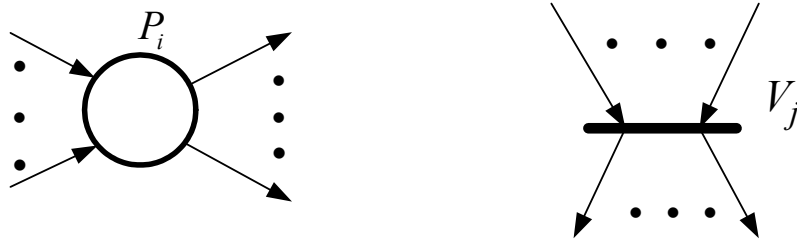


Рисунок 3.6 – Графічне позначення позицій P_i і переходів V_j

Місце й перехід можуть мати кілька вхідних і вихідних дуг, однак дуги повинні з'єднувати тільки вершини різних типів – позиції P_i й переходи V_j .

Розглянемо довільну мережу Петрі (рис. 3.7).

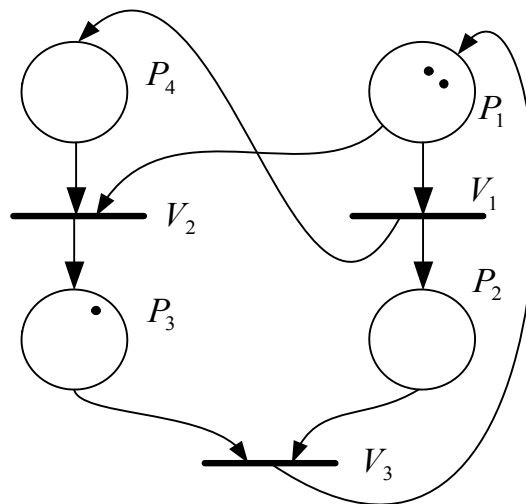


Рисунок 3.7 – Довільна мережа Петрі

Позиції P_1 й P_3 містять точки. Точки називаються фішками або мітками. При зміні станів вони переміщуються по мережі. Якщо фішок багато, вони можуть бути позначені цифрами. Одній фішці відповідає або один з n паралельних процесів, що містять відповідні дуги, або один з операторів одного процесу.

Подія або дія відбуваються в переході при наявності умов. Перехід можна представити як логічний елемент «I» з декількома входами. Перехід моделює оператори. Установка фішок у мережі називається розміткою мережі, а переміщення фішок відбувається тільки при спрацьовуванні переходу, тобто при настанні події.

У мережі Петрі місце (позиція) моделює умови. Наявність умови визначається наявністю фішки. Дуга може бути задана предикатом (вираженням «так-ні» або логічною функцією.

Коли фішки, що рухаються по дузі, пересунуться в цю позицію, інформація по дузі, що виходить з позиції поступить на перехід, у якому реалізується оператор, тобто відбудеться сама подія.

Таким чином, функціонування мережі можна трактувати як послідовність дискретних подій.

3.5 Формальне визначення мережі Петрі

Формально мережа N визначається п'ятіркою множин:

$$N = \langle P, V, F, H, M_0 \rangle,$$

де $P = \langle P_1, P_2, \dots, P_n \rangle$ або $\forall P_i \in P$ – кінцева непуста множина символів P_i , названих місцями (позиціями) мережі;

$V = \langle V_1, V_2, \dots, V_m \rangle$ або $\forall V_i \in V$ – кінцева непуста множина символів V_i , названих переходами;

F – функція інцидентності (табл. 3.1), що вказує на наявність дуг, які з'єднують місця P_i з переходами V_j , причому, якщо $F(P_i, V_j) = 1$, то така дуга є, а якщо $F(P_i, V_j) = 0$, такої дуги немає;

H – функція інцидентності (табл. 3.2), що вказує на наявність дуг, які з'єднують переходи V_i з місцями P_j , причому, якщо $H(V_i, P_j) = 1$, така дуга є, а якщо $H(V_i, P_j) = 0$, такої дуги немає;

M_0 – початкова розмітка мережі Петрі, що представляє собою множина місць у безлічі цілих позитивних чисел $P \rightarrow \{0, 1, 2, \dots\}$, які вказують кількість фішок на кожному місці.

Таблиця 3.1 – Функція інцидентності F

F	V_1	V_2	V_3
P_1	1	1	0
P_2	0	0	1
P_3	0	0	1
P_4	0	1	0

Таблиця 3.2 – Функція інцидентності H

H	P_1	P_2	P_3	P_4
V_1	0	1	0	1
V_2	0	0	1	0
V_3	1	0	0	0

Наприклад, для наведеної вище мережі:

$P = \{P_1, P_2, P_3, P_4\}$ – множина місць;

$V = \{V_1, V_2, V_3\}$ – множина переходів;

Початкова розмітка:

$$M_0 = \frac{P_1}{2} \mid \frac{P_2}{0} \mid \frac{P_3}{1} \mid \frac{P_4}{0}.$$

Аналіз початкової розмітки M_0 й функції інцидентності F , представлених у таблиці, дозволяє визначити, що спрацює перехід V_1 , з'єднаний з вершиною (місцем) P_1 , яка містить дві фішки. Інші переходи не спрацьовують, тому що для спрацьовування V_2 потрібна наявність фішки в P_4 , а для спрацьовування V_3 – в P_2 . Після спрацьовування V_1 утворюється нова розмітка M_1 :

$$M_1 = \frac{P_1}{1} \mid \frac{P_2}{1} \mid \frac{P_3}{0} \mid \frac{P_4}{1},$$

яка показує, що до спрацьовування готові всі три переходи. Враховуючи те, що при програмній реалізації одним процесором дві події не можуть

відбутися одночасно, то далі аналіз будується на основі випадкового вибору.

Нехай, наприклад, спрацював перехід V_2 і утворювалася розмітка M_2 :

$$M_2 = \frac{P_1}{0} \left| \frac{P_2}{1} \right| \frac{P_3}{2} \left| \frac{P_4}{0} \right|$$

Переходячи від одного вектора розмітки до іншого, можна записати ланцюжок спрацювування переходів:

$$M_0 \left| \frac{V_1}{M_1} \right| \frac{V_2}{M_2} \left| \frac{V_3}{M_3} \right| \frac{V_1}{M_4} \left| \frac{V_3}{M_5} \right| \frac{V_1}{V_2} \Rightarrow \frac{M_6}{M_7}$$

Розмітки $M_6 = \{0,1,1,2\}$ й $M_7 = \{0,0,1,0\}$ з'явилися тупиковими, тому що при таких розташуваннях фішок жоден перехід не спрацює й мережа зависає.

3.6 Застосування мереж Петрі для моделювання

При проектуванні нових і дослідженні існуючих дискретних систем керування застосовується кілька способів, які укладаються у два підходи.

Перший підхід полягає в тому, що проєктована система спочатку представляється традиційним способом (графом, блок-схемою алгоритму), потім вона моделюється мережею Петрі, аналіз якої дозволяє виявити вади й модернізувати мережу.

Другий підхід протилежний першому. Проектування починається зі створення мережі Петрі, її дослідження й звільнення від вад, а потім по мережі будується традиційний алгоритм.

При реалізації першого підходу для правильного перекладу блок-схеми алгоритму в мережу Петрі необхідно кожен дугу вихідної програми представити вершиною-місцем у мережі Петрі, а вузли блок-схеми (аналіз умов, розв'язок, введення, вивід і т.д.) замінити переходами (рис. 3.8).

Для успішного застосування мереж Петрі з метою моделювання програм потрібна різна інтерпретація умов і позицій (місць), подій і переходів. Мережі Петрі можуть також застосовуватися для моделювання апаратних засобів від простих логічних пристроїв до процесорів і ЕОМ.

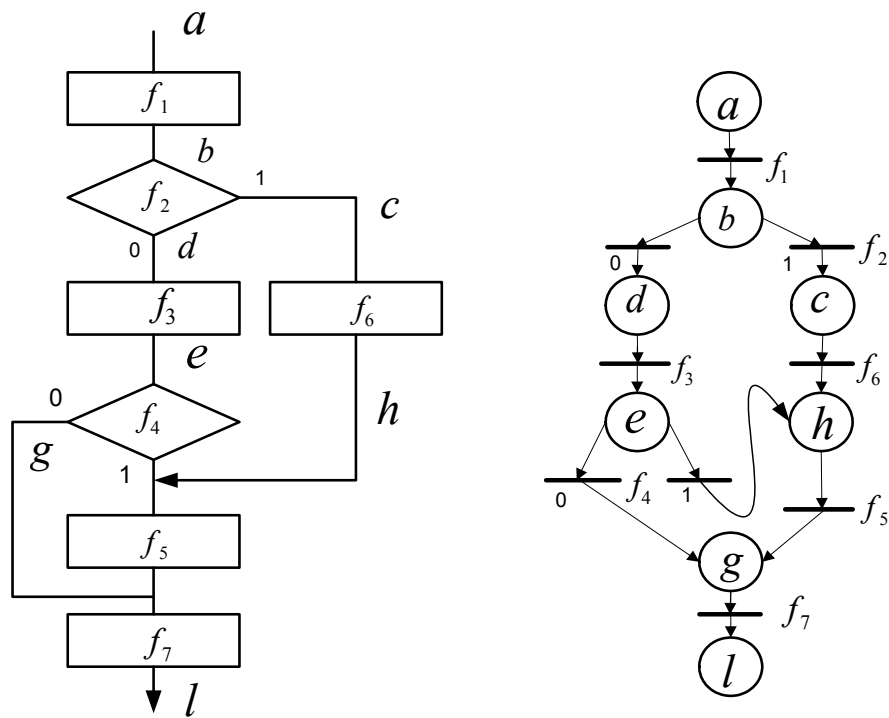


Рисунок 3.8 – Приклад перекладу блок-схеми алгоритму в мережу Петрі

Якщо моделюється кінцевий автомат, то початкова розмітка задається однією фішкою, що поміщається в початковий стан автомата. Кожному стану автомата зіставляється місце в мережі Петрі, а кожна дуга розрізається переходом і позначається тим же символом, яким позначена дуга на графові переходів автомата.

3.7 Розробка мережі Петрі для моделювання циклу автоматичної зміни інструмента

Розглянемо процес автоматичної зміни інструмента на токарському верстаті із ЧПУ. На токарських верстатах інструментальні магазини виконують у вигляді револьверної головки, у гніздах (позиціях) якої розміщуються від 6 до 12 інструментів.

Зміна інструмента проводиться шляхом обертання головки навколо осі, розташованої в горизонтальній або вертикальній площині. Напрямок повороту визначається шляхом вибору меншої величини кута між вихідною й заданою позицією. При виконанні повороту головка розкріплюється, а по завершенню повороту фіксується й затискається. Для контролю всіх операцій установлені датчики – кінцеві вимикачі в механізмі затискання й розтискання, шляхові датчики позицій, кінцевий вимикач механізму фіксації.

Цикл ініціюється введенням команди T<двухзначне число>, а починається при подачі команди M06 і наявності сигналів з інших автоматичних ву-

злів, що дозволяють роботу, наприклад: “робоча зона закрыта захисним кожухом”, “вісь повороту головки вилучена від інших вузлів верстата” і ін.

Автоматичний цикл складається із трьох паралельних процесів:

- А – приймання Т-команди й М-команди, запуск циклу, індикація позиції;
- В – керування процесом затискання й розтискання revolverної головки;
- С – керування процесом повороту головки й пошук заданої позиції.

У кожному процесі проводиться аналіз умов, формування керуючих сигналів і реалізація подій.

Цикл автоматичної зміни інструмента може бути змодельований мережею Петрі, наведеною на рисунку 3.9.

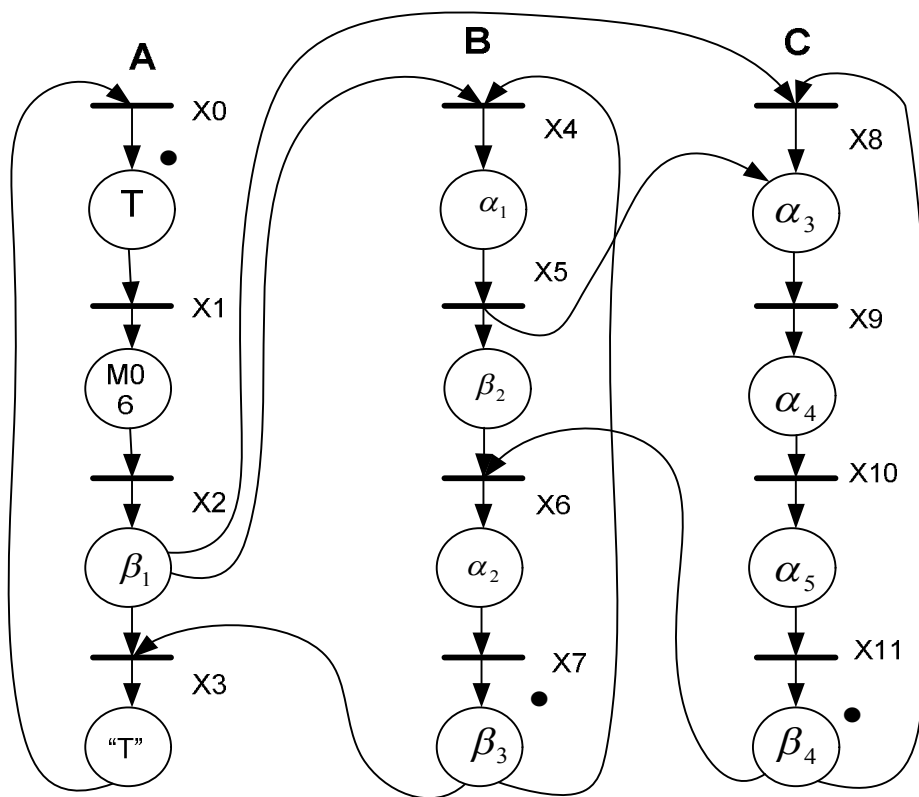


Рисунок 3.9 – Мережа Петрі для циклу автоматичної зміни інструмента

Визначимо умови (місця) і події (переходи) у кожному процесі.

Процес А починається з уведення команди Т, яка вказує код позиції інструмента. Процес аналізує код завдання й фактично встановлену позицію, тобто визначає необхідність запуску циклу зміни інструменту.

Далі в переході x_1 формується сигнал «Цикл заданий» і фішка переміщується в місце очікування команди М06 (Зміна інструменту).

Після команди М06 спрацьовує перехід x_2 і фішка переміщується в міс-

це β_1 з умовами виконання циклу:

- номер позиції інструмента заданий;
- блокування запуску процесу В немає.

Умови для спрацьовування переходу x_3 , що формує сигнали на панель про виконання Т-команди, настануть після реалізації процесу В.

У процесі В при наявності умов «голівка затиснута» і «запуск процесу В дозволений» спрацьовує перехід x_4 і фішка переходить у місце α_1 , де аналізується наявність команди розтиску голівки. При поданні цієї команди спрацьовує перехід x_5 , фішка переміщається в місце β_2 , де приймається сигнал з кінцевого вимикача розтиску та формуються умови для включення процесу С.

Процес С починається з аналізу умов для спрацьовування переходу x_8 :

- голівка фіксована;
- код позиції введений.

При збігу умов (фішки в місцях β_1 , β_4) перехід x_8 спрацьовує й фішка процесу С переміщається в місце α_3 з умовами для аналізу напрямку повороту. Вибір найближчого шляху приводить до спрацьовування переходу x_9 і фішка переміщається в місце α_4 аналізу позицій, зчитуваних зі шляхових датчиків. При виході в задану позицію спрацьовує перехід x_{10} , фішка переміщається в місце α_5 аналізу сигналу кінцевого вимикача про фіксацію голівки. По цьому сигналу спрацьовує перехід x_{11} і фішка переходить у місце β_4 , у яким формується сигнал дозволу затискання голівки, тобто продовження процесу В.

Наявність фішок у β_2 й β_4 приводить до спрацьовування переходу x_6 і виконанню операції затискання, фішка процесу В переходить у місце α_2 очікування сигналу «голівка затиснута», який свідчить про закінчення процесу В. Після цього фішка переміщається через перехід x_7 у місце β_3 включення індикації встановленого інструмента.

3.8 Моделювання процесу керування гнучкими виробничими модулями (ГВМ)

З погляду керування ГВМ є більш складним об'єктом, чим верстат із ЧПУ. Він містить у собі (крім верстата) робот, накопичувач деталей, магазин заготовок, магазин інструментів, транспортну систему, а також систему автоматичного контролю.

Усі ці компоненти ГВМ постачені власними засобами керування, які поєднуються в єдину локальну керуючу мережу. Взаємодія всіх засобів керування здійснюється за допомогою програми диспетчера. Диспетчеризація – це основна функція СУ ГВМ, що забезпечує автоматизацію процесу виготовлення деталей.

Особливість ГВМ полягає в його використанні для обробки однотипних виробів з різним змістом операцій, набором застосовуваних інструментів, геометричними параметрами. Тому СУ ГВМ повинна вирішувати наступні завдання:

- 1) визначати по спеціальних ознаках технологічний процес обробки деталі й необхідний для цього інструмент, тобто вирішувати завдання ідентифікації;
- 2) спостерігати за ресурсом роботи інструмента, діагностувати відхилення розмірів оброблюваних поверхонь і вводити корекцію, ухвалювати розв'язки для виходу з позаштатних ситуацій, тобто вирішувати завдання моніторингу;
- 3) сприймати команди вищого рівня, вести діалог з оператором, передавати інформацію про хід виконання завдань і ін., тобто вирішувати термінальні завдання.

Для забезпечення взаємодії об'єктів ГВМ розробляється програма диспетчера. При розробці цієї програми необхідно:

- 1) визначити склад паралельних процесів управління, їх апаратні структури, інформаційні й виконавчі пристрої;
- 2) розділити кожний процес на окремі управляючі програми, скласти специфікації програм;
- 3) виділити умови виконання кожної програми в рамках ГВМ, визначити необхідні для цього обмінні й блокувальні сигнали, розробити засоби для їхньої передачі;
- 4) розробити алгоритм аналізу умов і виклику програм.

Розглянемо спрощену структуру ГВМ у складі верстата, робота й магазину для заготовок і деталей (рис. 3.10).

Кожна одиниця встаткування має власний пристрій програмного керування й можливості об'єднання в мережу.

СУ магазину шляхом повороту й фіксації магазину забезпечує подачу заготовки в позицію x_0 , а також прийом до вільного місця (гнізда) обробленої деталі.

СУ робота дозволяє переміщати захват в початковий стан x_1 захоплювати заготовку в зоні x_0 , переносити її в позицію x_2 для обробки заготовки й виводити захват в безпечну зону x_3 , управляти процесом затискання й розтискання захвата.

СУ верстата забезпечує закріплення, розкріплення й обробку заготовки.

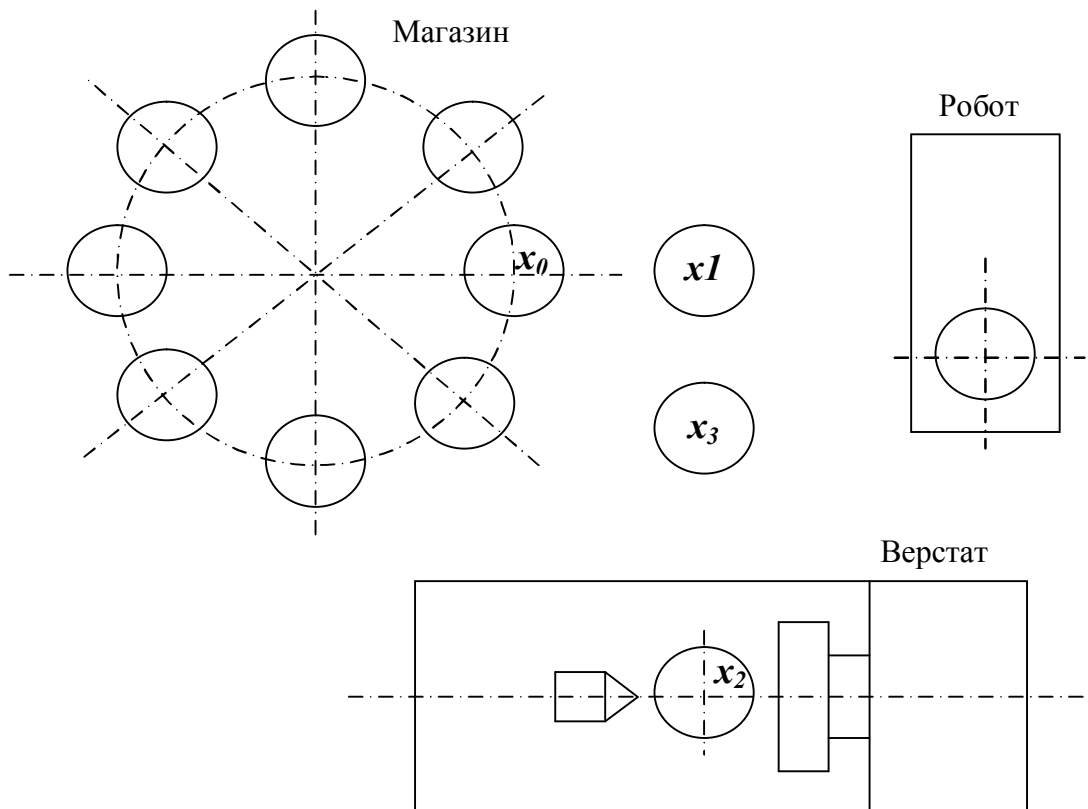


Рисунок 3.10 – Спрощена структура ГВМ

Специфікація програм і їх умовні позначки представлено в таблиці 3.3.

Процес взаємодії програм вимагає обліку наступних умов:

- 1) початкова установка всіх СУ може здійснюватися одночасно;
- 2) спочатку диспетчер повинен дозволити виконання програми M1 (подача заготовки в зону x_0);
- 3) до початку обробки деталі виконується програма P1 (захват заготовки в магазині і її переміщення в зону верстата x_2), потім B1 (затискання заготовки на верстаті), після цього захват виводиться в безпечну зону x_3 програмою P2;
- 4) після обробки деталі (програма B2) дозволяється виконання про-

грам у наступній послідовності: P3 (захват деталі), B2 (розтискання заготовки), M2 (підготовка місця для приймання деталі) і P4 (установка деталі в магазин і вихід у зону x_1);

5) якщо є завдання на обробку наступної деталі, диспетчер виводить СУ в стан початкової установки, а якщо ні, то виконуються M3, P5 і B4, тобто проводиться перепрограмування, настроювання й зміна інструмента.

Таблиця 3.3 – Специфікація й позначення управляючих програм

Система управління	№ п/п	Зміст програми	Умовна позначка
Магазин	1	Початкова установка (стан готовності)	M0
	2	Подача заготовки в зону x_0	M1
	3	Подача в зону x_0 вільного місця під деталь	M2
	4	Поворот у початковий стан для завантаження й вивантаження магазину	M3
Робот	1	Початкова установка, переміщення захвата в зону x_1	P0
	2	Захват заготовки в зоні x_0 і переміщення її в зону обробки x_2 через безпечну зону x_3	P1
	3	Розтиск захвата, переміщення захвата в безпечну зону x_3	P2
	4	Переміщення захвата в зону x_2 , захват (затиск) деталі	P3
	5	Переміщення захвата в зону x_0 , розтиск захвата, вихід у позицію x_1	P4
	6	Вихід у стан перепрограмування або очікування	P5
Верстат	1	Початкова установка верстата	B0
	2	Затискання заготовки	B1
	3	Обробка заготовки	B2
	4	Розтискання заготовки	B3
	5	Вихід у стан перепрограмування або очікування	B4

Мережа Петрі для процесу керування ГВМ показана на рисунку 3.11.

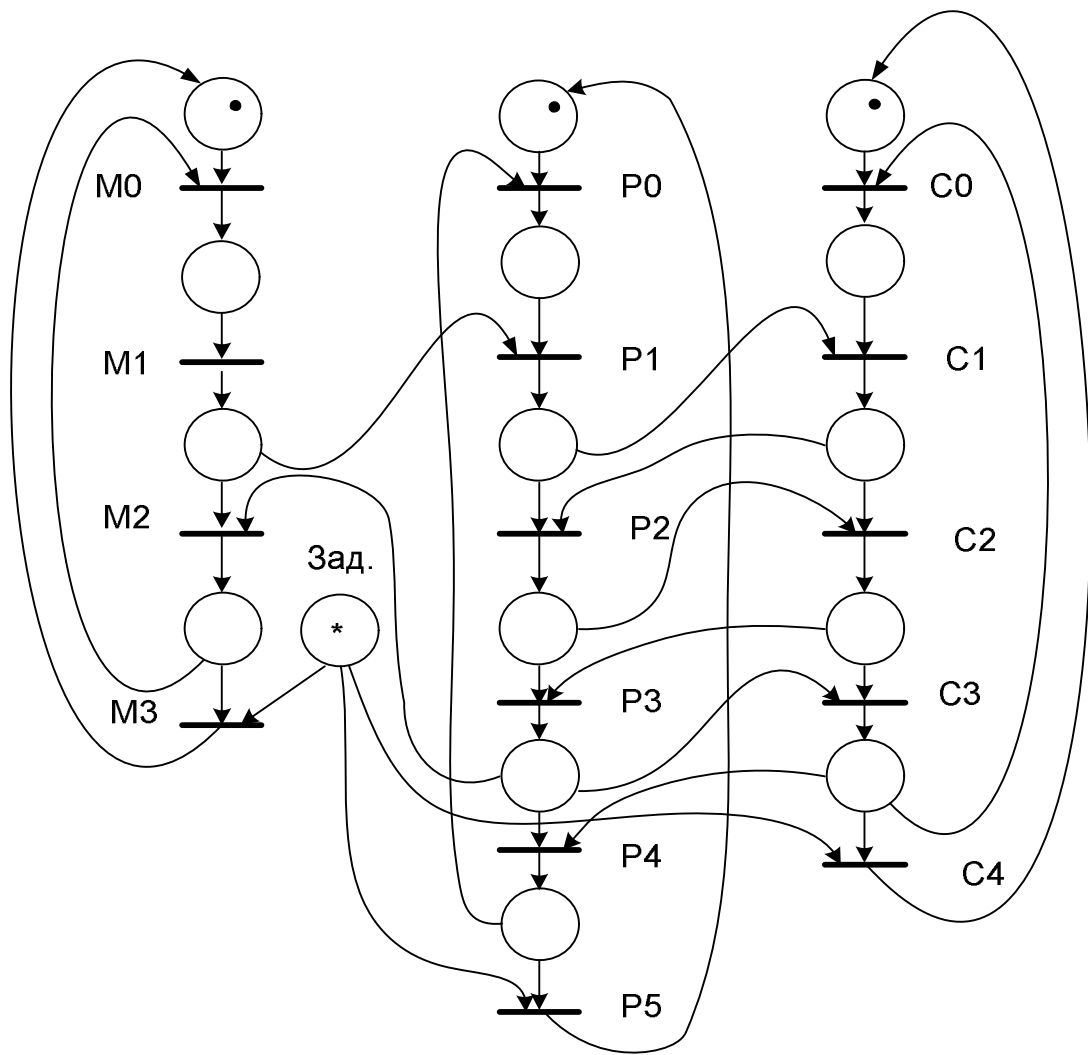


Рисунок 3.11 – Мережа Петрі для процесу керування ГВМ

4 МОДЕРНІЗАЦІЯ СИСТЕМ ЧПУ

4.1 Аналіз цілей і завдань модернізації

Модернізації підлягає встаткування, яке експлуатувалося довгий час. Користувачі верстатів усе частіше й частіше зважаються на модернізацію електроніки, коли механічні деталі й вузли верстата перебувають у гарному стані.

Мета проекту модернізації полягає в підвищенні ефективності використання верстата.

При модернізації верстата, користувачі можуть одержати вигоду залежно від завдань реконструкції:

- Поліпшення обслуговування верстата.
- Прискорення робочих циклів.
- Підвищення точності.
- Збільшення обсягу програмного забезпечення й розширення функцій системи керування.
- Можливість обробки більш складних поверхонь і деталей.
- Інтеграцію системи ЧПУ в інформаційний простір підприємства.

Модернізації підлягають наступні компоненти систем ЧПУ:

- засоби керування й організація інтерфейсу оператора.
- інформаційні й виконавчі пристрої;
- пристрої перетворення сигналів;
- алгоритмічне й програмне забезпечення.

При розробці проекту модернізації, як методологічний напрямок повинен бути прийнятий ***системний підхід***, який дозволяє розкрити цілісність процесу проектування в умовах різноманіття типів зв'язків, а також об'єднати цифрові й аналогові канали, механічні й електронні пристрої, програмні й апаратні засоби.

Проектування повинне базуватися на технологічному процесі. Технологічними критеріями функціонування об'єкта є точність підтримки технологічних параметрів і режимів, продуктивність процесу, надійність і гнучкість системи керування.

Завданнями проектування є пошук технічних розв'язків, що поліпшують технологічні критерії.

Найбільш складним завданням проектування є пошук розв'язків, що забезпечують підвищення точності обробки деталей на верстаті.

Точність залежить практично від усіх компонентів системи керування:

- зазорів і сил тертя в кінематичних ланках;
- місця установки, статичних і динамічних погрішностей датчиків;

- погрішностей розрахунків;
- пружних відхилень інструмента й деталі в статичних і динамічних режимах, впливи зовнішніх збурювань;
- структури й параметрів регулятора;
- нелінійностей у каналах керування й зворотному зв'язку.

Завдання підвищення точності повинне вирішуватися шляхом ретельного аналізу механізмів формування погрішностей і розробки комплексних заходів, спрямованих на наступне:

- скорочення довжини кінематичного ланцюга між робочим органом і датчиком положення;
- підвищення чутливості вимірювальних перетворювачів;
- вибір оптимальної структури системи керування;
- вибір оптимального алгоритму цифрового регулятора;
- уведення програмної або апаратної компенсації нелінійностей ланок цифрового електропривода;
- забезпечення необхідної швидкодії й обчислювальної потужності керуючого контролера.

Слід урахувати, що ступінь підвищення точності роботи верстата досить складно зв'язати з результатами проектування. Тому розрахунки економічної ефективності таких проектів викликають значні проблеми.

Іншим завданням проектування, яке визначається вимогами виробництва, є завдання підвищення продуктивності встаткування.

Продуктивність – це показник, який визначається як величина, зворотна сумі втрат часу на здійснення робочих рухів, виконання допоміжних операцій, а також на виконання ремонтних і інших робіт. Тому розв'язок завдання підвищення продуктивності вимагає зменшення втрат часу.

Для зменшення втрат часу необхідно, насамперед, збільшити надійність роботи всіх компонентів системи – електроніки, механіки, гідравліки й т.п.

Надійність – властивість об'єкта зберігати в часі у встановлених межах значення всіх параметрів, що характеризують здатність виконувати необхідні функції в заданих режимах і умовах застосування, технічного обслуговування, зберігання й транспортування.

Інтуїтивно надійність об'єктів зв'язують із неприпустимістю відмов у роботі. Це є розуміння надійності в «вузькому» змісті – властивість об'єкта зберігати працездатний стан протягом деякого часу або деякого наробітку. Інакше кажучи, надійність об'єкта полягає у відсутності непередбачених неприпустимих змін його якості в процесі експлуатації й зберігання. Надійність тісно пов'язана з різними сторонами процесу експлуатації.

Надійність в «широкому» змісті — комплексна властивість, яка залежно від призначення об'єкта й умов його експлуатації може містити в собі власти-

вості *безвідмовності, довговічності*, ремонтпридатності й збереженості, а також певна комбінація цих властивостей.

Збільшення періодів безвідмовності й довговічності прямо пов'язане зі зменшенням часу простоїв на обслуговування й ремонт, а отже, зі збільшенням продуктивності.

Найбільш ефективними засобами підвищення надійності є:

- вибір елементів, які мають найменшу ймовірність відмови;
- проектування засобів захисту від аварій;
- розробка розвинутої системи діагностики.

Продуктивність устаткування пов'язана із ще одним показником – коефіцієнтом технічного використання.

Коефіцієнт технічного використання – показник, що характеризує ступінь *продуктивного* використання встаткування. Розраховується за часом з урахуванням потужності (продуктивності) і обсягу зробленої продукції або виконаної роботи. Коефіцієнт використання встаткування за часом визначається шляхом розподілу часу фактичної роботи встаткування на плановий фонд часу, тобто на кількість годин роботи встаткування, передбачену планом з урахуванням числа календарних днів у періоді, святкових і вихідних, встановленого режиму роботи, тривалості зміни, а також часу на планово-запобіжний ремонт.

Якщо верстат повинен був працювати в даному місяці 160 годин, а практично через простої, не передбачені планом, працював 150 годин, то коефіцієнт використання встаткування за часом рівний 93,8% (6,2% – це втрати верстатного часу). Важливо забезпечити роботу встаткування не тільки без простоїв, але й із встановленою потужністю або продуктивністю.

Для підвищення коефіцієнта технічного використання встаткування необхідно прагнути до розширення функціональних можливостей системи ЧПУ й верстата, забезпечити універсальність застосування й швидку переналадку.

Важливим завданням модернізації верстата є також зниження витрат на виготовлення одиниці продукції, тобто **зменшення собівартості продукції**. Мова може йти, наприклад, про економію енергетичних ресурсів за рахунок використання ефективних систем приводів або про економію фінансових витрат на використання обслуговуючого персоналу.

У проекті модернізації потрібно прагнути до того, щоб якість продукції й продуктивність верстата не залежали від обслуговуючого персоналу. Однак повністю виключити можливість ручного керування верстатом не можна.

Для задоволення вимог погодження взаємодій оператора, устаткування й системи керування повинні бути передбачені:

- робота в ручному, налагоджувальному й автоматичному режимах;

- виконання підрежимних функцій;
- зв'язок по послідовному каналу, засоби для мережної роботи, підключення до промислової мережі Ethernet;
- засоби керування приводами й автоматикою.

Вибір режиму керування повинен здійснюватися оператором. Автоматичний режим є основним (робочим) режимом. Ручний режим передбачений для проведення ремонтно-налагоджувальних робіт, ліквідації аварійних і по-заштатних ситуацій, а також для використання верстата без системи програмного керування. Відмінною рисою цього режиму є те, що всі сигнали керування впливають безпосередньо на відповідні виконавчі пристрої, минаючи локальну систему автоматичного керування.

При автоматизованому режимі керування оператор повинен мати можливість здійснювати ручне втручання в роботу системи для коректування програми.

Усі функції керування й захисту приводів повинні бути інтегровані в перетворювачі частоти. Перетворювачі частоти повинні забезпечувати наступні (основні) функції захисту:

- від короткого замикання;
- від перевантаження;
- від підвищення напруги в ланцюзі постійного струму;
- від зниження напруги в ланцюзі постійного струму;
- від обриву фази живлячої напруги;
- від перегріву радіаторів силових елементів;
- від перегріву двигуна;
- від замикань на землю силових ланцюгів;
- від збою в мікропроцесорній системі.

Спрацьовування будь-якого захисту повинне приводити до відключення електропривода. Інформація про спрацьовування захисту повинна передаватися на дисплей панелі керування.

Усі сигнали керування повинні відповідати стандартам і підтримувати нормальне функціонування у випадку заміни компонентів системи. Система ЧПУ повинна забезпечувати:

- формування дискретних сигналів постійного струму $U=24\text{ В}$, $I=0,2\text{ А}$ на реле й $I=5\text{ А}$ на потужні електромагнітні системи;
- вивід сигналів змінного струму $U=110\text{ В}$, $I=2\text{ А}$ на пускачі й іншу апаратуру;
- вивід аналогових сигналів через ЦАП $U=\pm 10\text{ В}$, $I=20\text{ мА}$ на регулятори (точність $\delta =1\%$);
- вивід синусоїдальних сигналів для живлення датчиків індуктивного типу (відхилення по амплітуді $0,1\%$);
- уведення аналогових сигналів напруги й струму з датчиків;

- уведення дискретних сигналів $U=24\text{ В}$ постійного струму з подорожніх і кінцевих вимикачів, а також сигналів змінного струму $U_{\sim}=110\text{ В}$ від кнопок пульта керування встаткуванням.

Для забезпечення погодженої роботи системи ЧПУ й верстата повинне бути зроблене *статичне, динамічне й програмне* узгодження.

Статичне узгодження – це забезпечення відповідності граничних рівнів сигналів керування граничним значенням параметрів об'єкта в статичному режимі (min, max). Із цією метою у пам'ять системи записуються константи: граничні значення переміщень по координатах, діапазони швидкостей рухів і частот обігу, передаточні числа редукторів і датчиків зворотного зв'язку, ціна дискрети переміщення по кожній координаті, образ пульта оператора, склад інформації, яка виводиться на дисплей і ін.

Динамічне узгодження – це обмеження змінних, які характеризують роботу об'єкта, а також забезпечення необхідної точності спостереження в динамічному режимі, здійснюване за допомогою констант (допустиме прискорення, максимальна швидкість, допустима помилка, таблиця коригувальних величин ходового гвинта, інші константи, які враховують особливості встаткування).

Програмне узгодження передбачає доробку прикладного програмного забезпечення системи – розробку програм керування автоматикою, а також програм регуляторів і обробки переривань.

4.2 Модернізація верстатів ЧПУ на базі систем SINUMERIK

Системи ЧПУ SINUMERIK фірми SIEMENS мають модульну конструкцію й великий набір компонентів для реалізації будь-якої конфігурації. Докладна інформація про параметри систем ЧПУ і їх основних компонентах наведена у каталозі NC 60, а також на сайті www.siemens.ua/mc.

Система містить у собі три базові комплекти, які відрізняються, головним чином, кількістю керованих осей і набором реалізованих функцій:

- SINUMERIK 802 с модифікаціями S, C, D.
- SINUMERIK 810D.
- SINUMERIK 840D.

Вказані системи ЧПУ постачені *модулями привода SIMODRIVE*, які розроблені спеціально для верстатів із ЧПУ. Передбачені також модулі позиціонування FM STEPDRIVE.

Основними компонентами керування в системах ЧПУ SINUMERIK є панель оператора OP, а також термінальні блоки NCU (Numerical Computer Unit), верстатні пульти MCP та електронні маховички. Процесорні модулі оснащені операційною системою Windows NT 4.0.

При виборі системи ЧПУ слід керуватися її призначенням.

Системи SINUMERIK 802 призначені для токарських, фрезерних, свердлильних і шліфувальних верстатів. Система SINUMERIK 802D являє собою один блок PCU (Panel Control Unit), призначений для розв'язку простих завдань – керування чотирма осями й одним шпинделем. Для керування автоматикою передбачений модуль PP 72/48, а для керування приводами – перетворювач SIMODRIVE 611 universal E із аналоговим каналом управління.

Приклад схеми підключення системи SINUMERIK 802C наведено на рисунку 4.1.

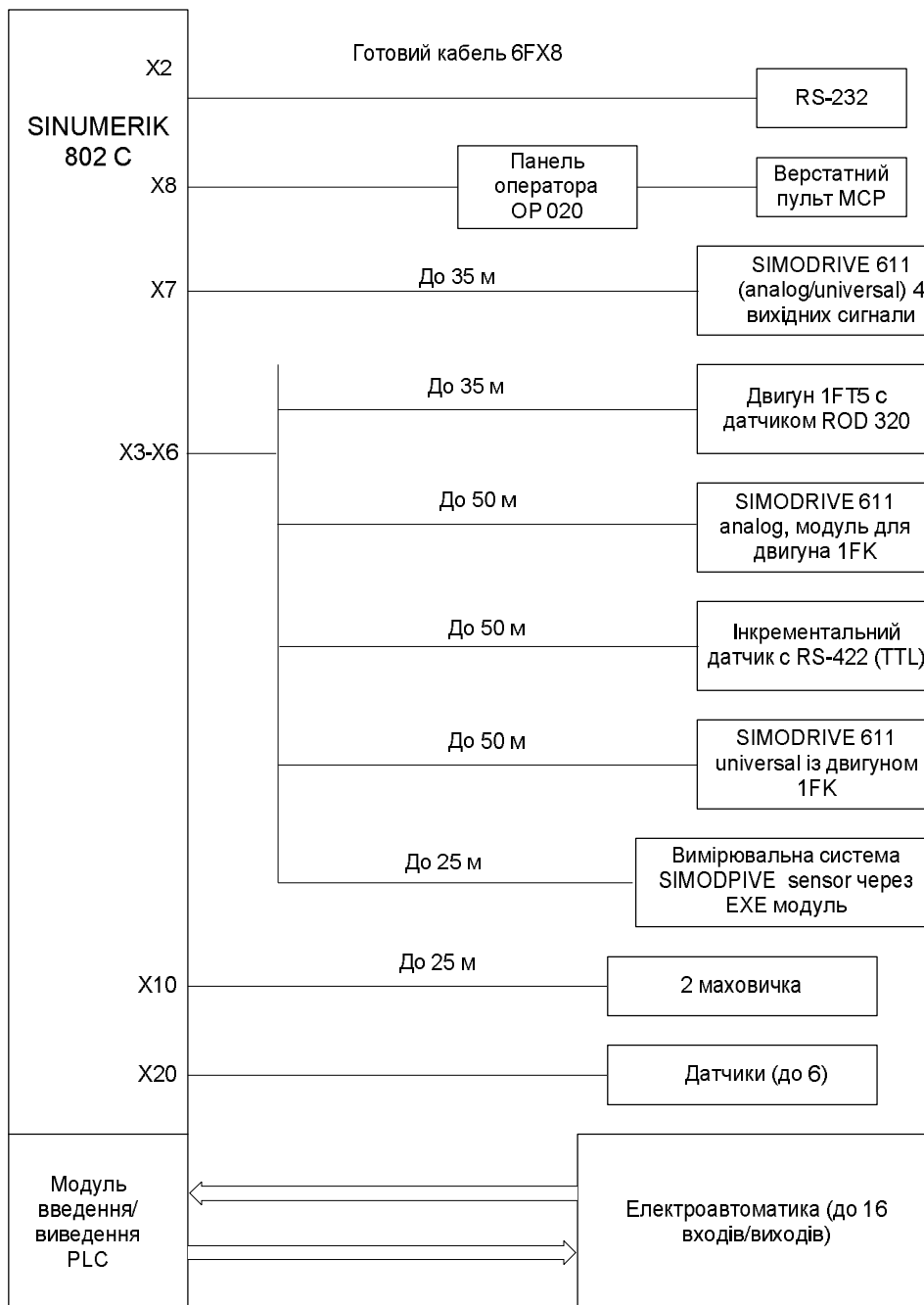


Рисунок 4.1 – Схема підключення SINUMERIK 802C

У системах SINUMERIK 802 кількість одночасно інтерпольованих осей не перевищує трьох. Привод шпинделя може управлятися по швидкості або положенню.

Обробка сигналів датчиків положення осей в SINUMERIK 802S/802C виконується в модулі системи ЧПУ, а в SINUMERIK 802D – у цифрових модулях привода SIMODRIVE 611 E (universal).

Обмеження руху по координатах забезпечується за допомогою апаратних і програмних кінцевих вимикачів. Апаратні безконтактні кінцеві вимикачі встановлюються для виключення аварій при роботі на прискореному русі й при русі з робочою подачею. Для зменшення зони переміщень можуть бути встановлені програмні вимикачі швидкого й робочого ходу. Передбачена також функція “Наїзд на жорсткий упор”, яка необхідна, наприклад, для фіксації задньої бабки при затисканні деталі в центрах.

Погрішність ходового гвинта або вимірвальної системи може бути компенсована по таблиці компенсацій, складеної за результатами вимірів фактичних відхилень.

Система SINUMERIK 810D поєднує в одному модулі CCU (Compact Control Unit) усі завдання ЧПУ, програмувального контролера й комунікації. CCU перебуває в одному корпусі з інтегрованими силовими модулями, що забезпечують роботу із двома приводами подачі й одним приводом шпинделя. З метою розширення кількості осей (максимально до 6) використовуються спеціальні плати розширення й перетворювачі SIMODRIVE 611 із цифровим інтерфейсом.

Системи SINUMERIK 840D – це системна платформа, на якій можна реалізувати найбільш складні завдання керування. У комбінації із програмувальним контролером SIMATIC S7-300 і перетворювачами SIMODRIVE 611 digital вона дозволяє створювати комплекси програмного керування із числом керованих осей від 2 до 31.

Приклад схеми підключення системи SINUMERIK 840D показано на рисунку 4.2.

Система побудована на базі багатопроцесорного модуля NCU (Numerical Control Unit), у якому об'єднані завдання числового програмного керування, логічного керування автоматикою (PLC) і комунікації. До цієї системи ЧПУ можуть бути підключені:

- Панель керування з PCU (Panel Control Unit) або модулем MMC (мікрокартою пам'яті), верстатний пульт, кнопкова панель і інші засоби ручного керування.
- Перетворювачі SIMODRIVE 611 digital.
- Програмактор.
- Периферія PLC SIMATIC S7-300.

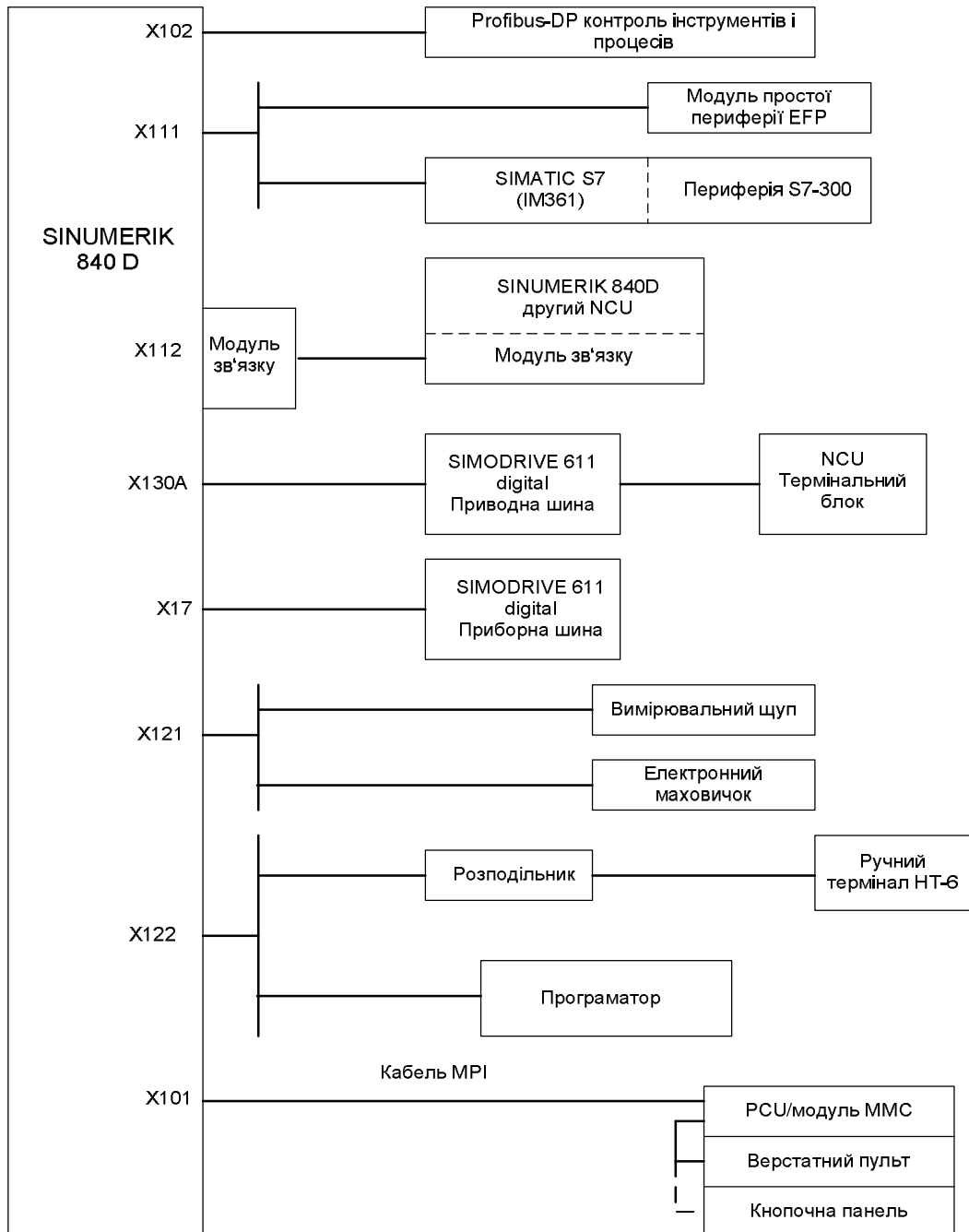


Рисунок 4.2 - Схема підключення SINUMERIK 840D

- Децентралізована периферія (станція ET200M) через шину PROFIBUS.
- Модуль простої периферії EFP (64 входи й 32 виходів на 24 В із клемними колодками).
- Термінальний блок NCU (один або два) для забезпечення безпеки робочих рухів на верстаті. Він підключається до приводної шини й у нього можна вставити до 8 компактних модулів введення/виводу DMP. Блок NCU з модулями DMP забезпечує швидке синхронне введення/вивід дискретних і аналогових сигналів системи безпеки. На кожний термінальний блок можна підключити до 16 цифрових

каналів уведення/виводу або до 4 каналів аналогових уведення/виводів.

Приводна система SIMODRIVE 611

Приводна система SIMODRIVE 611 виконана в єдиній конструкції по модульному принципу. Завдяки стандартним інтерфейсам і з'єднанням користувач може побудувати конфігурацію з будь-якою комбінацією координатних осей і шпинделів.

Приводна система складається з наступних компонентів:

1. Трансформатор (при необхідності узгодження напруг).
2. Мережний фільтр і комутуючий дросель для зниження рівня радіоперешкод, які створюють перетворювачі частоти.
3. Модуль живлення (нерегульований UE-модуль або регульований модуль живлення/рекуперації E/R).
4. Силові модулі (перетворювачі частоти для двигунів).
5. Плати керування (аналогові, цифрові й універсальні), настроєні на певні типи й технології використання двигунів.

Приводна система підключається до мережі із глухозаземленою нейтраллю (TN-мережа) напругою 400 В, або 415 В, або 480 В частотою 50/60 Гц через наступні пристрої: трансформатор (при необхідності), фільтр, комутуючий дросель, модуль живлення.

Модуль живлення виробляє напругу постійного струму 490 В або 680 В для проміжного контуру, а також напруги $\pm 24V$, $\pm 15V$, $+5V$ для електроніки.

Напруга проміжного контуру може бути нерегульованою або регульованою. Нерегульована напруга застосовується для приводів потужністю 5, 10 і 28 кВт із незначними динамічними навантаженнями. Регульована напруга від модулів E/R застосовується для приводів потужністю від 16 до 120 кВт у наступних випадках:

- високі динамічні вимоги до приводів верстатів;
- часті цикли гальмування й висока енергія гальмування;
- коли висуваються вимоги оптимізації експлуатаційних витрат.

За допомогою регульованих модулів живлення/рекуперації надлишкова енергія проміжного контуру, яка виникає, наприклад, у режимі гальмування, вертається в мережу. Таким чином отримуються оптимізовані параметри для охолодження розподільної шафи й рентабельний баланс енергії для користувача.

Кодовим перемикачем можна вибрати різні робочі функції модулів E/R:

1 Регульований режим на напрузі проміжного контуру 600 В DC або 625 В DC.

2 Нерегульований режим на напрузі проміжного контуру 490 В DC (при 3 АС 400 В) з або без мережної рекуперації.

3 Пряме використання мережі TN 3 AC 480 В +6 % -10 %; 50 Гц/60Гц на нерегульовану напругу проміжного контуру 680 В DC з мережною рекуперацією.

Зовнішній вигляд модуля живлення показаний на рисунку 4.3.



Рисунок 4.3 - Зовнішній вигляд модуля живлення для SIMODRIVE 611

Напруга проміжного контуру від силового модуля подається на всі силові модулі лінійки приводів SIMODRIVE 611.

Силові модулі управляються системою ЧПУ через плати керування. Цифрові сигнали керування передаються по приводній шині ЧПУ. Зовнішній вигляд силового модуля SIMODRIVE 611 показаний на рисунку 4.4.

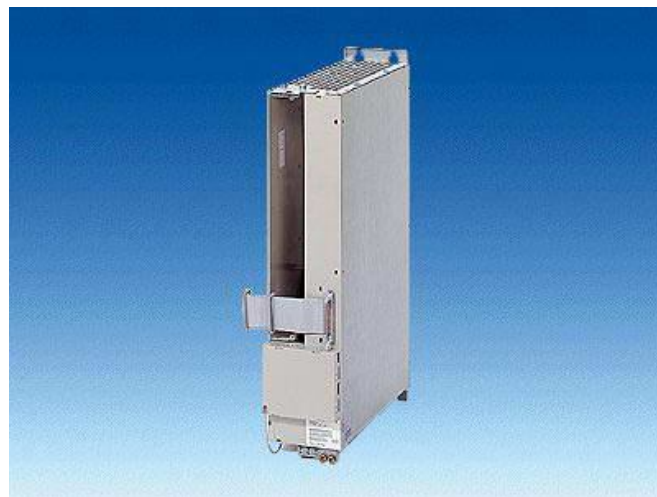


Рисунок 4.4 – Зовнішній вигляд силового модуля Simodrive 611 (плата керування знята)

Структурна схема приводної системи SIMODRIVE 611 показана на рисунку 4.5.

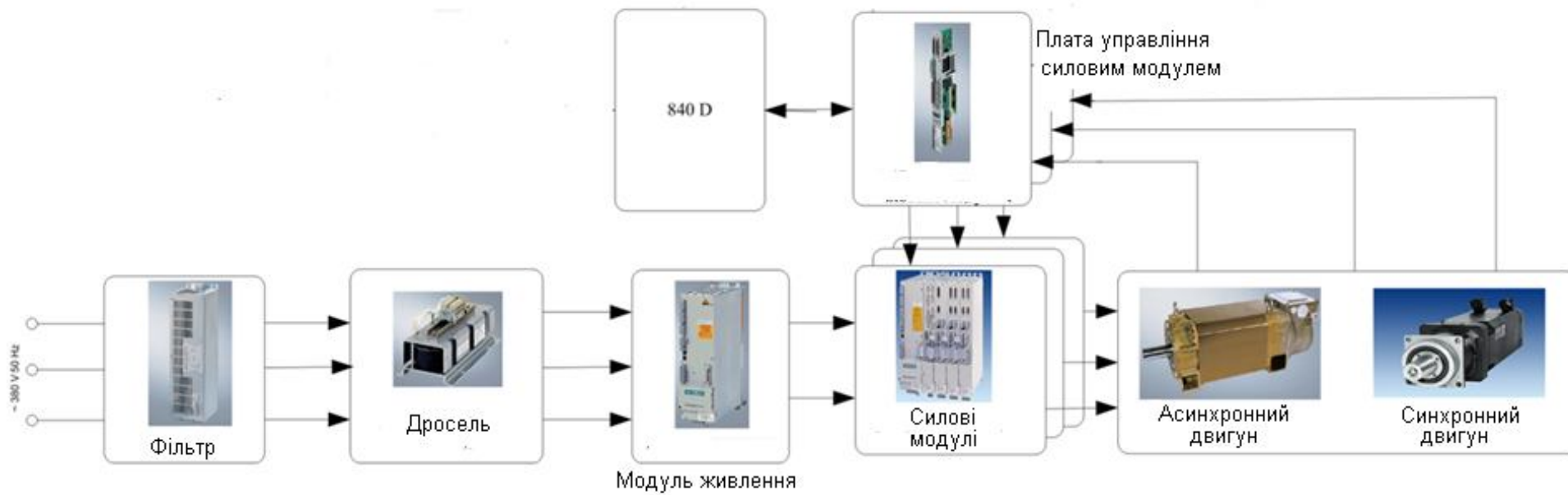


Рисунок 4.5 – Структурна схема приводної системи SIMODRIVE 611

Цифрові плати управління системи SIMODRIVE 611 вставляються в силові модулі й використовуються для керування трифазними серводвигунами 1FT6/1FK або лінійними двигунами 1FN приводів подачі, а також двигунами 1PM/1PH/1FE1 для привода шпинделя. До цифрових плат керування можна підключати датчики, вбудовані у двигуни, а також зовнішні вимірювальні системи, що видають сигнали \sin/\cos ,

Схема підключення силового модуля SIMODRIVE 611 і плати керування із цифровим інтерфейсом наведена на рисунку 4.6.

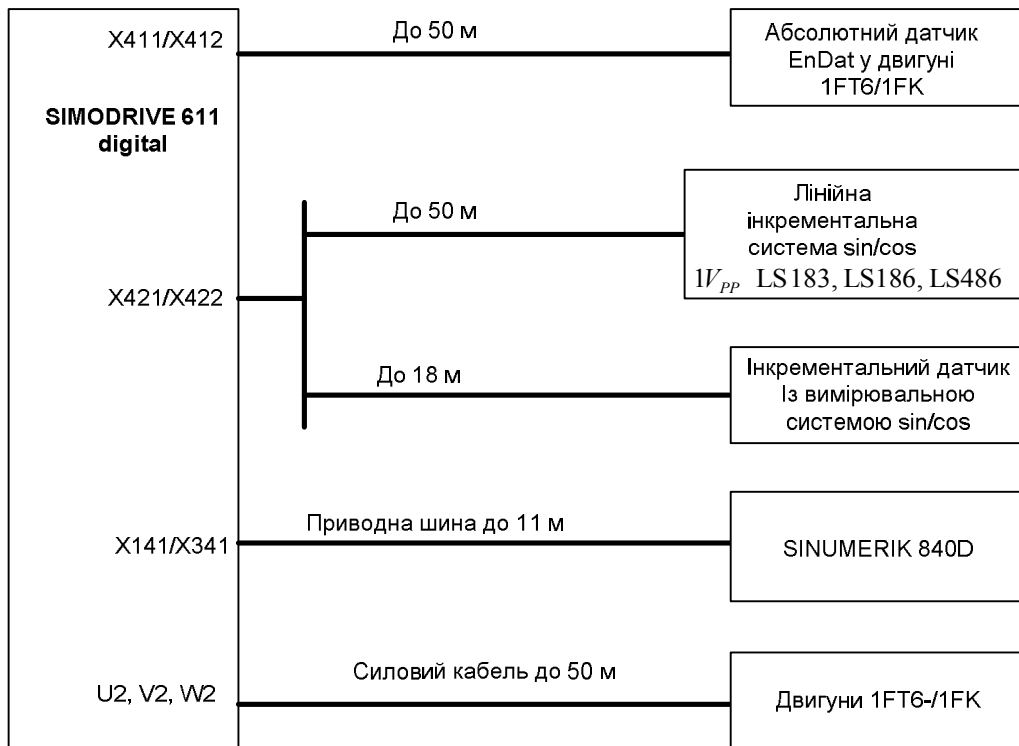


Рисунок 4.6 - Схема підключення перетворювача SIMODRIVE 611 digital

У двигуни 1FT6/1FK і 1PM/1PH можуть бути вбудовані інкрементальні фотоелектричні датчики з вихідними сигналами \sin/\cos напругою 1 В, а також абсолютні датчики Endat з живленням +5В, які виконуються однооборотними (13 розрядів) і багатооборотними (25 розрядів).

Для визначення положення рухливого вузла при невеликим переміщенні рекомендується використовувати вимірювальну лінійку HEIDENHAIN LB 382ур, яка дозволяє точно визначити положення робочого вузла. Зовнішній вигляд вимірювальної лінійки HEIDENHAIN LB 382ур представлено на рисунку 4.7.

Для визначення положення рухливого вузла при значнім його переміщенні доцільно використовувати вимірювальну лінійку HEIDENHAIN LIM 581, зовнішній вигляд якої представлено на рисунку 4.8.



Рисунок 4.7 - Зовнішній вигляд вимірювальної лінійки HEIDENHAIN LB 382up



Рисунок 4.8 - Зовнішній вигляд вимірювальної лінійки HEIDENHAIN LIM 581

Технічні характеристики вимірювальних лінійок LB 382up і LIM 581 наведені в таблицях 4.1 і 4.2, відповідно.

Таблиця 4.1 - Технічні характеристики вимірювальної лінійки HEIDENHAIN LB 382up

Модель	LB 382 up to ML 3040mm
Вимірювальний стандарт	Стрічка нержавіючої сталі із градацією AURODUR 40 μm
Розподілений інтервал	
Тепловий коефіцієнт розширення	$\alpha_{\text{therm}} = 10 \text{ ppm/K}$
Градус точності	$\pm 5 \mu\text{m}$ ($\pm 0.0002 \text{ in.}$)
Вимірювальна довжина	до 3040 мм
Максимальна швидкість переміщення	120 м/хв
Живлення	5 V \pm 5% / < 150 mA

Таблиця 4.2 - Технічні характеристики вимірювальної лінійки HEIDENHAIN LIM 581

Модель	LIM 581
Вимірювальний стандарт	Магнітна пластикна куля на сталевій стрічці
Розподілений інтервал	10,24 μm
Тепловий коефіцієнт розширення	$\alpha_{\text{therm}} = 10 \text{ ppm/K}$
Точність	$\pm 10 \mu\text{m}$ ($\pm 0.0002 \text{ in.}$)
Вимірювальна довжина	до 28000 мм
Максимальна швидкість переміщення	600 м/хв
Живлення	5 V \pm 5% / < 150 mA

4.3 Розробка структурної схеми системи ЧПУ верстата і її конфігурування

При розробці структурної схеми системи керування верстата важливо показати, які саме технічні засоби і їх модифікації використані в цій системі.

Приклад вистави структурної схеми представлений на рис. 4.9.

До складу цієї системи керування входять:

- система ЧПУ SINUMERIK 840D;
- програмувальний контролер автоматки SIMATIC S7-300 із процесором CPU-314C і станцією ET200M із модулями введення-виводу для розподіленої периферії;
- панель оператора 12" TFT з рідкокристалічним екраном;
- комплект вимірювальних лінійок, що забезпечують точність відліку не гірше 0,001мм;
- комплект модулів цифрового керування приводами подач типу SIMODRIVE 611;
- комплект високомоментних сервомоторів для приводів подач;
- перетворювач SIMOREG для приводу головного руху із двигуном постійного струму;
- комплект засобів для ручного керування.

Порядок проектування структурної схеми системи керування полягає в наступному.

Об'єкт керування розділяється на ряд конструктивних вузлів і механізмів. У результаті декомпозиції повинні бути отримані структурні компоненти, які включають у себе один об'єкт керування, наприклад, привод подачі.

Після цього здійснюється аналіз сигналів, необхідних для керування кожним структурним компонентом.

Для аналізу вхідних і вихідних дискретних сигналів їх слід розділити по функціональному призначенню. У наведених нижче таблицях сигнали розділені на наступні групи (приклад):

- вхідні сигнали, відповідальні за живлення системи (табл. 4.3);
- вхідні сигнали керування елементами гідравтоматики (табл. 4.4);
- вхідні сигнали, відповідальні за позиціонування й роботу системи керування приводом подачі супорта (табл. 4.5).
- вихідні сигнали контролера, які ставляться до керування приводами подачі (табл. 4.6).

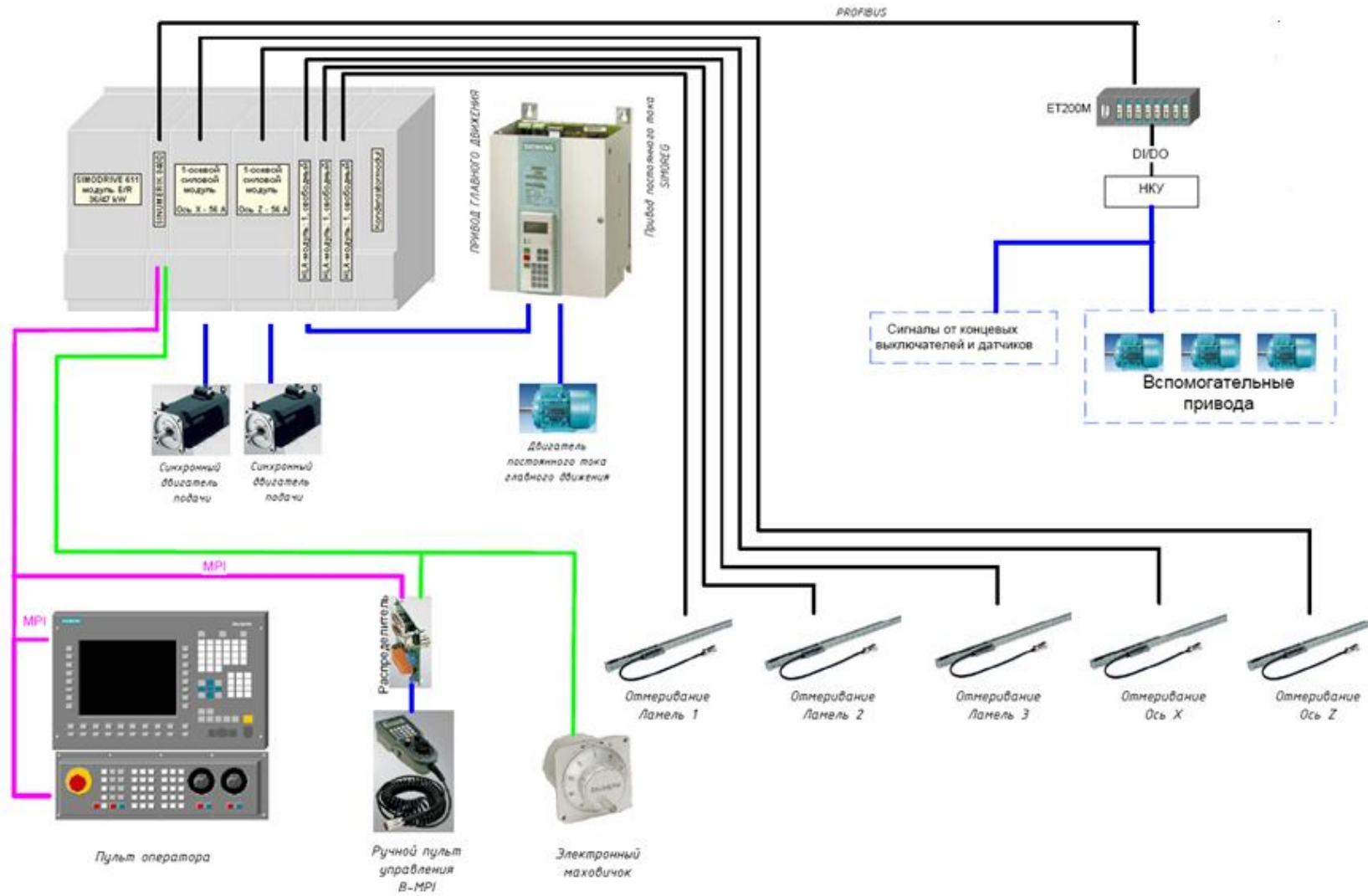


Рисунок 4.9 – Структурна схема системи керування токарського верстата HOESCH DB3000

Таблиця 4.3 - Вхідні дискретні сигнали, відповідальні за живлення (приклад, фрагмент)

Символ	Тип	Призначення
AC 220V 1 Ok	BOOL	220В, 50Гц у нормі
AC 24V Ok	BOOL	24В, 50Гц у нормі
Operating_supply_24V Ok	BOOL	+24У в нормі
Power_supply_24V Ok	BOOL	Силова напруга 24В в нормі
Infeed_modul Ok	BOOL	Модуль живлення в нормі
Infeed_module_no	BOOL	Модуль живлення виключений
Infeed_moduel_contact_ON	BOOL	Внутрішній контактор модуля живлення включений
AC 220V main cut out Ok	BOOL	Вхідний автомат включений
Power_supply_ETA 24V Ok	BOOL	Силова напруга 24В в нормі
No OFF machine	BOOL	Вимикання верстата немає
Machine_ON_1	BOOL	Верстат включений
Machine_ON_2	BOOL	Верстат включений
Check_Fuse_4	BOOL	Контроль запобіжників групи 4
Check_Fuse_5	BOOL	Контроль запобіжників групи 5

Таблиця 4.4 - Вхідні дискретні сигнали гідравтоматики керування приводом подачі супорта (приклад, фрагмент)

Символ	Тип	Призначення
ST_Protect_Pump_lubr	BOOL	Автомат двигуна насоса проточного змащення осі X, Z
ST_Flow_axis_Z_OK	BOOL	Недолік протоки вісь Z
ST_Flow_axis_X_OK	BOOL	Недолік протоки вісь X
ST_Protect_Pump1_X	BOOL	Автомат двигуна маслонасоса 1 гідростатики
ST_Protect_Pump2_X	BOOL	Автомат двигуна маслонасоса 2 гідростатики X
ST_Flow_X1_OK	BOOL	Недолік протоки маслонасоса 1 гідростатики X
ST_Flow_X2_OK	BOOL	Недолік протоки маслонасоса 2 гідростатики X
ST_Protect_Pump1_Z	BOOL	Автомат двигуна маслонасоса 1 гідростатики Z
ST_Pump1_ON_OK	BOOL	Контроль включення контактора насоса 1 гідростатичного черв'яка

Таблиця 4.5 - Вхідні дискретні сигнали, відповідальні за роботу системи керування приводом подачі (приклад, фрагмент)

Символ	Тип	Призначення
Taster_15	BOOL	Квітування користувацьких помилок
No_Emergency_stop_1	BOOL	Немає аварійного останова 1
No_Emergency_stop_2	BOOL	Немає аварійного останова 2
HHU_ON	BOOL	Виносний пульт включений
HHU_enable_1	BOOL	Дозволена робота виносному пульту
ST_Limit_switch_-X	BOOL	Обмеження ходу осі -X
ST_Limit_switch_+X	BOOL	Обмеження ходу осі +X
ST_Reference_switch_X	BOOL	Прив'язка осі X
No_Emergency_stop_9	BOOL	Немає аварійного останова 9
ST_Limit_switch_-Z	BOOL	Обмеження ходу осі -Z
ST_Limit_switch_+Z	BOOL	Обмеження ходу осі +Z
ST_Reference_switch_Z	BOOL	Прив'язка осі Z

Таблиця 4.6 – Вихідні сигнали контролера SIMATIC S7-300 (приклад, фрагмент)

Символ	Тип	Призначення
ST_Pump_ON	BOOL	Включення насоса гідростатичного розвантаження черв'яка
ST_Pump_prior_ON	BOOL	Включення насоса попереднього тиску
ST_prior_pressure_ON	BOOL	Включення електромагніту попереднього тиску
ST_Pump_recycle_ON	BOOL	Включення насоса, що відсмоктує
ST_cooling_fluid_ON	BOOL	Включення електромагніту охолоджуючого засобу
ST_air_ON	BOOL	Включення електромагніту повітря
Main_circuit_ON_enable	BOOL	Дозвіл включення верстата

Після визначення кількості й атрибутів (властивостей) інформаційних і керуючих сигналів проводиться вибір базової системи, наприклад, SINUMERIK 810D, і її компонування необхідними засобами керування. Ця процедура називається конфігуруванням системи керування.

В існуючій системі ринку діють спеціальні програми, які дозволяють автоматизувати процес вибору апаратних засобів і проектування конфігурацій. Така програма розроблена й для систем SINUMERIK відомого німецького концерну SIEMENS.

Для конфігурування системи ЧПУ SINUMERIK фірма SIEMENS пропонує застосовувати конфігуратор NCSD, який являє собою універсальну програму по інтерактивному каталогу продукції фірми. При запуску програми встановлюється меню для вибору засобів і “дерево” проекту.

Процес проектування починається з вибору базового комплекту – SINUMERIK 802, SINUMERIK 810 або SINUMERIK 840. Далі з урахуванням необхідної продуктивності вибирається тип центрального процесора ЧПУ. Перевірка обраного процесора здійснюється по кількості керованих осей координат.

У базовій конфігурації система комплектується NCU 561.4. Цей пристрій ЧПУ може виконувати тільки прості завдання керування по двом координатам – лінійна вісь та шпиндель. При цьому лінійна вісь виконує тільки позиціонування, а інтерполяція реалізується у вигляді зв'язку двох осей (для нарізування різьблення). Для забезпечення керування більш складними процесами потрібно буде встановити інший модуль NCU, вибравши його з наведених на вкладці “NCU” (рис. 4.10).

Вибір NCU проводиться з урахуванням кількості необхідних груп режимів роботи (ГРР), кількості каналів програмування й кількості підтримуваних осей. Слід урахувати, що нормальна конфігурація передбачає одну групу режимів роботи (ГРР). У пропонованій на рисунку 4.10 таблиці в дужках наведена максимальна кількість груп, які можуть бути створені в цій системі.

NCU	Опции ЧПУ 1		Опции ЧПУ 2	
	ГРР	Канали	Осей	
<input checked="" type="radio"/> Нет NCU				
<input type="radio"/> NCU 561.4	1	1 (2)	1 (2)	
<input type="radio"/> NCU 561.5	1	5 (6)	5 (6)	
<input type="radio"/> NCU 571.4	1	5 (6)	5 (6)	
<input type="radio"/> NCU 571.5	1 (10)	1 (10)	5 (31)	
<input type="radio"/> NCU 572.4	1 (10)	1 (10)	5 (31)	
<input type="radio"/> NCU 572.5	1 (10)	1 (10)	5 (31)	
<input type="radio"/> NCU 573.4	1 (10)	1 (10)	5 (31)	
<input type="radio"/> NCU 573.5	1 (10)	5 (31)	5 (31)	

Рисунок 4.10 – Вид вкладки “NCU” для вибору пристрою числового керування

Одна група режимів роботи поєднує канали ЧПУ осями й шпинделями в один оброблюваний блок, тобто являє собою багатоканальний пристрій ЧПУ. У середині групи кожна вісь може програмуватися в будь-якому каналі. Якщо канал один, а осей 5, то керування осями проводиться послідовно в одному робочому циклі.

Після вибору процесора вибирається панель оператора. При цьому враховуються вимоги до характеру й змісту інформації, яка повинна виводитися

операторові й уводиться оператором. Система може бути постачена різними панелями. Одна з панелей показана на рисунку 4.11.



Рисунок 4.11 - Зовнішній вигляд панелі оператора OP 012

Додатково до панелі оператора можна вибрати верстатний пульт, кнопкові панелі й переносні термінали, електронні маховички для ручного керування координатними переміщеннями, а потім здійснити вибір програмного забезпечення для організації людино-машинного інтерфейсу.

Після вибору панелі оператора й засобів ручного керування здійснюється конфігурування апаратури для керування автоматикою.

На цьому етапі спочатку потрібно визначити тип центрального процесора контролера. Для вибору пропонується три типи – CPU 314, CPU 315-2DP і CPU 317-2DP. Вони відрізняються швидкодією, обсягом робочої пам'яті, у якій виконується користувацька програма, і кількістю підтримуваних каналів вводу-виводу. Тому, приступаючи до вибору CPU, необхідно мати уявлення про зразковий обсяг користувацької програми, а також про кількість вхідних і вихідних сигналів (аналогових і дискретних). Кількість вхідних і вихідних сигналів необхідно ввести у відповідних полях на вкладці « SM-модулі» у вікні «Входи/Виходи». Замість завдання кількості входів-виходів можна вибрати кількість модулів (вікно «Список модулів вх/вых»). Якщо вхідні й вихідні пристрої вилучені від головної шафи керування, слід передбачити станції вилученого введення-виводу, наприклад, станцію ET200M.

Далі можна вибрати функціональні модулі FM353 і FM354. Функціональні модулі FM використовуються для керування кроковими двигунами (FM 353) і серводвигунами (FM 354) у завданнях позиціонування. При цьому кроковий двигун повинен управлятися через силовий модуль FM STEPDRIVE, а серводвигун – через інтерфейс (10В аналогового перетворювача SIMODRIVE 611. У випадку вибору модуля FM 353 необхідно буде вказати тип модуля STEPDRIVE, що залежить від крутного моменту крокового двигуна.

Найважливішим етапом процесу конфігурування є вибір електродвигунів для верстата. Двигуни представлені рядом каталогів: серводвигуни 1FT,

1FK, лінійні двигуни 1FN, крокові двигуни 1FL, двигуни для привода головного руху (шпинделя) 1PM, 1PH, 1FE.

Для вибору двигуна конфігуратор надає користувачеві їх опис, технічні дані й характеристики, конструктивне виконання, наявність гальма, ступінь захисту й напрямок виводу кабелю. Тут же можна вибрати модулі живлення, управляючі модулі й перетворювачі.

На заключному етапі вибираються монтажні комплекти, клемні блоки й рознімання для монтажу системи. В останньому пункті проекту вибирається необхідна документація й програмне забезпечення.

Приклад конструктивної вистави сконфігурованої системи ЧПУ SINUMERIK 810D наведено на рисунку 4.12.

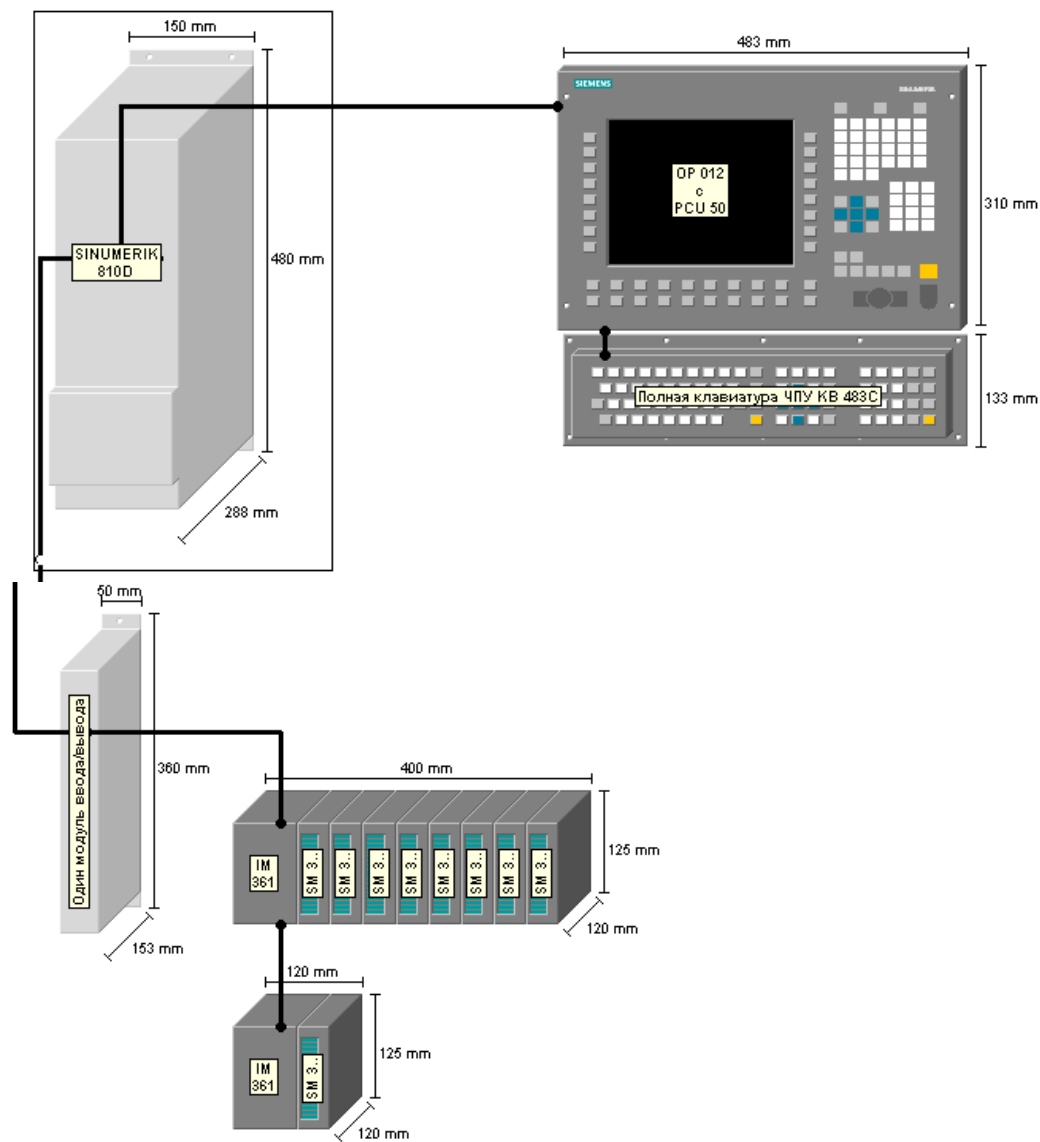


Рисунок 4.12 – Конструктивна вистава проекту системи ЧПУ SINUMERIK 810D

Пристрій ЧПУ SINUMERIK 810D (рис. 4.12) монтується в стійці перет-

ворювача SIMODRIVE 611, як показано на рисунку 4.13. Ця стійка, а також модуль простої периферії EFP, контролер SIMATIC S7-300, що полягає із двох стійок, з'єднаних інтерфейсними модулями IM 361, можуть бути розташовані в шафі керування. При цьому панель оператора із клавіатурою доцільно буде змонтувати безпосередньо на верстаті.

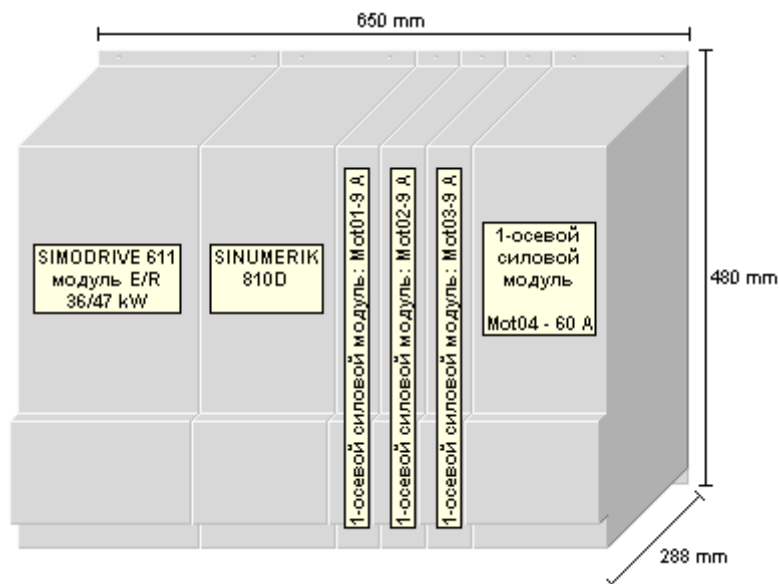


Рисунок 4.13 – Конструкція модулів SIMODRIVE 611

4.4 Розробка алгоритмів програмного забезпечення

Приклад блок-схеми головного модуля програми керування верстатом представлено на рисунку 4.14.

Перед селекцією режиму підпрограма SYSINIT здійснює ініціалізацію системи керування. При ініціалізації виконуються наступні дії: перевірка готовності до роботи силової частини приводів і електроавтоматики, установка робочих органів у нульове положення, скидання системи керування приводами.

Після ініціалізації очікується дозвіл на початок роботи, а якщо дозволу немає, програма завершується видачею відповідного повідомлення.

Режим автоматичної обробки починається з підпрограми PRGIN, яка забезпечує введення програми обробки із клавіатури або зчитування із деякого носія, наприклад, флешки. Програма вводиться покадрово, кожний кадр відповідає якій-небудь технологічній операції, наприклад, включенню шпинделя або обробці контуру із заданими координатами.

Для перетворення вхідної програми у внутрішній формат даних системи використовується інтерпретатор TRANS. Після інтерпретації даних введеного кадру подається запит на початок обробки. Якщо робота дозволена, починається виконання кадру з перевіркою закінчення програми (EOP).

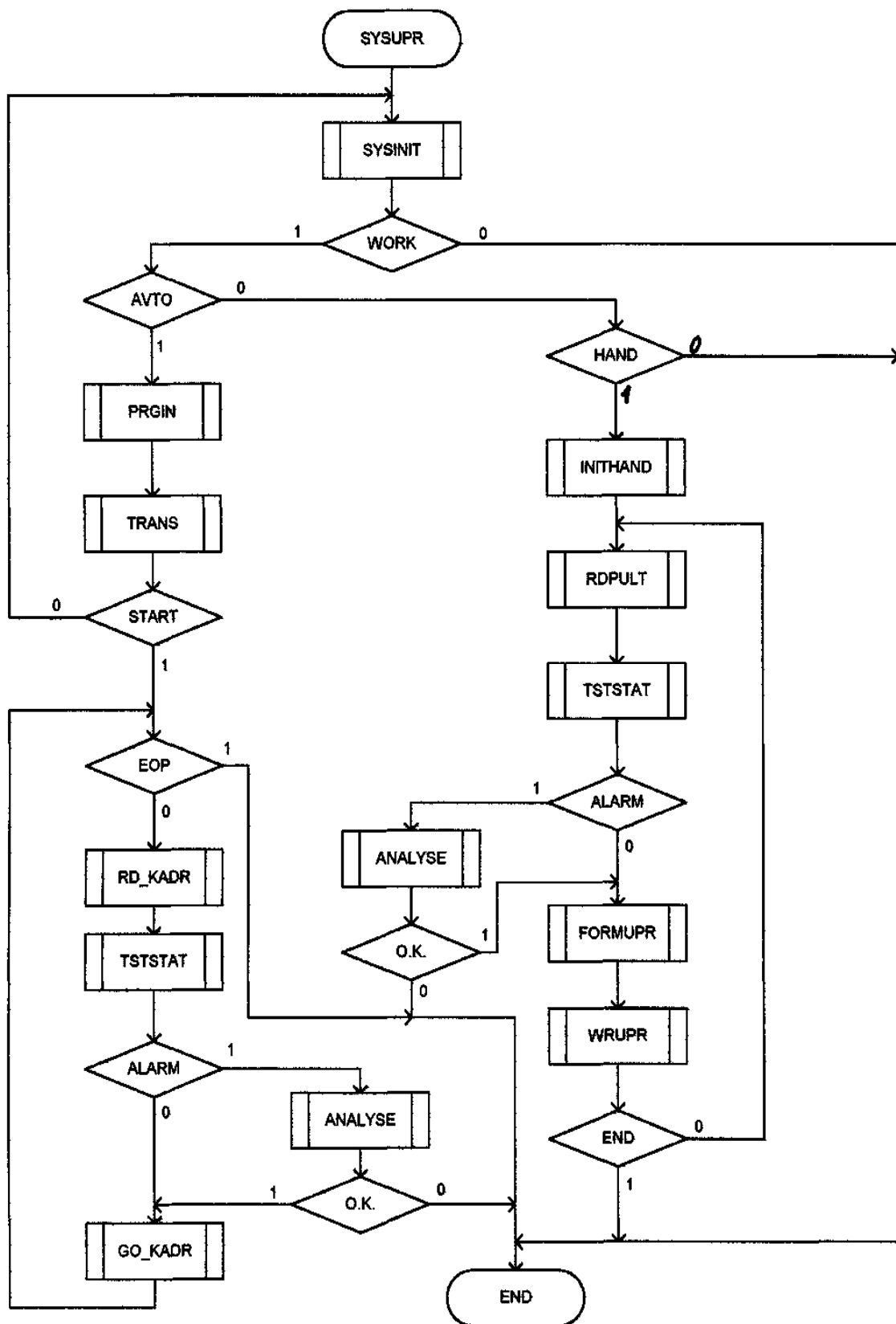


Рисунок 4.14 – Блок-схема головного модуля програми керування верстатом

Якщо програма не закінчена, виконується зчитування наступного кадру керуючої програми RD_KADR і його аналіз (визначення типу інтерполяції, робочого квадранта, величини переміщення, напрямку осей і т.д.), тобто підготовляється інформація для підпрограм нижнього рівня (інтерполятора й регулятора).

Перед відпрацюванням кадру виконується аналіз аварійних і блокувальних сигналів підпрограмою TSTSTAT. У випадку виникнення аварійної ситуації (ALARM=1) аналізується можливість її усунення, що виконується підпрограмою ANALYSE. Після завершення операції виходу з аварії керування передається підпрограмі обробки кадру керуючої програми GO_KADR.

Ручний режим роботи починається з виконання підпрограми INITHAND (рис. 4.14). Далі виконується читання пульта оператора (RDPULT). Тестування здійснюється підпрограмою TSTSTAT, яка виконує перевірку працездатності системи ЧПУ.

При відсутності блокувань відбувається формування керуючих слів підпрограмою FORMUPR, які визначають напрямок, код швидкості руху й величину переміщення по координаті, а також необхідні сигнали керування електроавтоматикою.

Дані слова записуються у відповідні регістри системи керування (підпрограма WRUPR). По закінченню режиму ручної обробки видається відповідне повідомлення й програма завершує свою роботу.

При розробці програми *контролера* необхідно передбачити аналіз користувачьких помилок, які виникають при керуванні верстатом. Компанія Siemens розробила стандартні програмні блоки для обробки стандартних помилок. Однак для конкретного застосування програміст може доповнити список помилок для зручності знаходження причин неполадки встаткування.

5 ЗАГАЛЬНА ХАРАКТЕРИСТИКА СТРУКТУРИ ТА КОМПОНЕНТІВ ПРИВІДНОЇ СИСТЕМИ SIMODRIVE

Привод SIMODRIVE має модульну структуру, у яку входять:

1. Двигун з датчиком положення й швидкості.
2. Привідний модуль (силовий модуль із платою керування).
3. Система живлення.

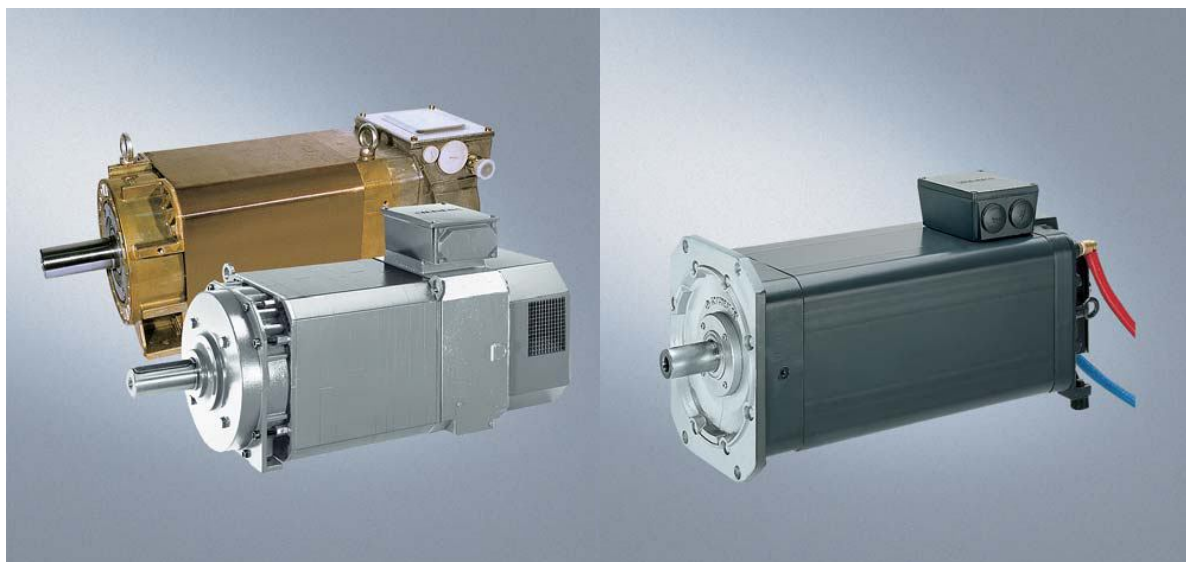
5.1 Загальна характеристика двигунів

За умовами роботи й вимогам до точності у верстатах із ЧПУ розрізняють наступні приводи:

1. Привод шпинделя (HSA).
2. Привод подачі (VSA).
3. Привод допоміжного механізму.

Для приводів шпинделя (головного руху) застосовують:

- асинхронні серводвигуни типу 1PH7 з повітряним охолодженням (рис. 5.1,а) і 1PH4 з водяним охолодженням (рис. 5.1,б);
- двигуни для прямого привода з водяним охолодженням 1PH2 (рис. 5.2);
- двигуни з порожнім валом 1PM4 водяного охолодження (рис. 5.3,а) і 1PM6 повітряного охолодження (рис. 5.3,б).



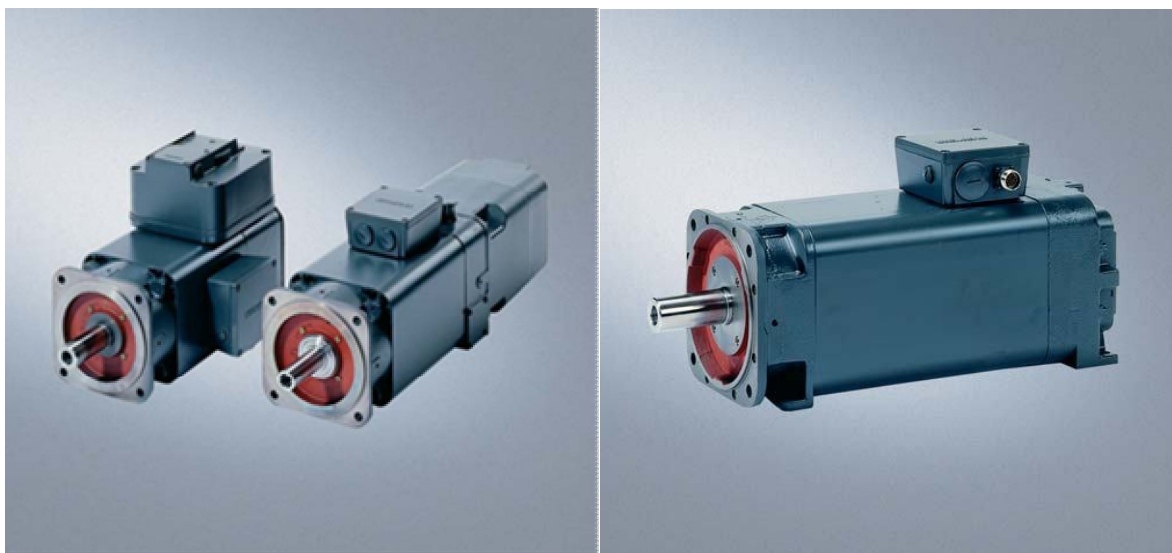
а

б

Рисунок 5.1 – Загальний вид двигуна 1PH7 (а) і двигуна 1PH4 (б)



Рисунок 5.2 – Загальний вид двигуна 1PH2



а

б

Рисунок 5.3 – Загальний вид двигуна 1PM4 (а) і двигуна 1PM6 (б)

Для приводів подачі застосовуються:

- синхронні серводвигуни 1FT5 із самоохолодженням або із примусовою вентиляцією, а також 1FT6 з водяним охолодженням (рис. 5.4);
- синхронні двигуни 1FK6 із самоохолодженням (рис. 5.5).

Для допоміжних приводів механізмів зміни інструментів, а також настановних переміщень, наприклад, переміщення задньої бабки токарського верстата, застосовують стандартні асинхронні двигуни без вбудованих датчиків контролю переміщень.

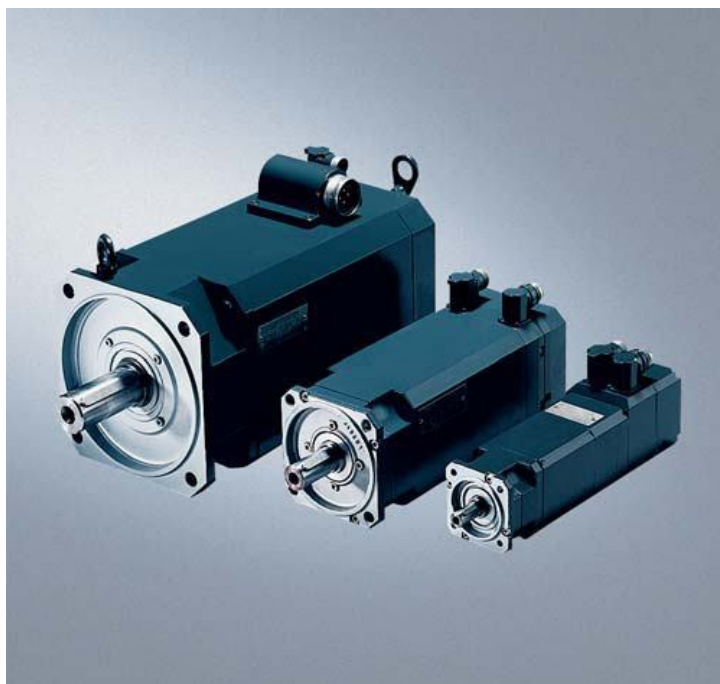


Рисунок 5.4 – Загальний вид двигуна IFT6



Рисунок 5.5 – Загальний вид двигуна IFK6

5.2 Огляд датчиків

Розрізняють інкрементальний і абсолютний методи виміру. При використанні інкрементальних датчиків після кожного відключення живлення необхідне реферування верстата, тому що положення зазвичай не зберігається в ЧПУ й рухи верстата при відключенні живлення не реєструються.

Абсолютні датчики, навпаки, реєструють рухи при відключенні жив-

лення й видають актуальну позицію при відновленні живлення, тобто ці датчики не вимагають реферування.

На рисунку 5.6 показані варіанти конструкцій ротаційних (кругових) датчиків і елементи для їхнього підключення.



Рисунок 5.6 – Датчики для приводів SIMODRIVE

У двигунах *головного руху* застосовуються різні типи датчиків

1. Інкрементальні кругові датчики. До цього типу ставляться:

- Датчики положення й швидкості шпинделя з імпульсними сигналами TTL з рівнем сигналу 5 В та HTL (High Voltage) з рівнем сигналу до 30 В.
- Датчики кута повороту з аналоговими сигналами sin/cos рівня напруги 1 Vpp (Volt peak to peak: напруга від піка до піка).
- Вимірвальна інкрементальна система датчика типу SIMAG H для реєстрації кута повороту порожніх шпинделів (двигуни 1PH2, 1FE1), що складається із шестірні й зчитувальної головки (заміняє колишню версію SIZAG 2).

2. Датчики абсолютного значення (кодові датчики положення). До цього типу ставляться:

- Датчики із синхронним послідовним інтерфейсом (SSI) (передача сигналів датчика проводиться під управлінням контролера, що не завжди можливо).
- Датчики із кодером по даним (інтерфейс Endat).
- Датчики з інтерфейсом PROFIBUS-DP.

3. Датчики кутового положення типу резольвер (resolver).

Інкрементальні й кодові датчики є *енкодерами* в тому розумінні, що їх

імпульсні сигнали повинні перетворюватися в N-код.

В інкрементальних датчиках, що містять синусний (А) і косинусний (В) канали, а також канал R для референтної мітки початку оберту, порядок чергування імпульсів дозволяє визначити напрямок обертання, підрахунок імпульсів у лічильниках – відстань, а визначення числа імпульсів за одиницю часу – швидкість. Діаграми сигналів TTL і аналогових сигналів sin/cos рівня напруги $1 V_{pp}$ показано на рисунках 5.7 і 5.8, відповідно.

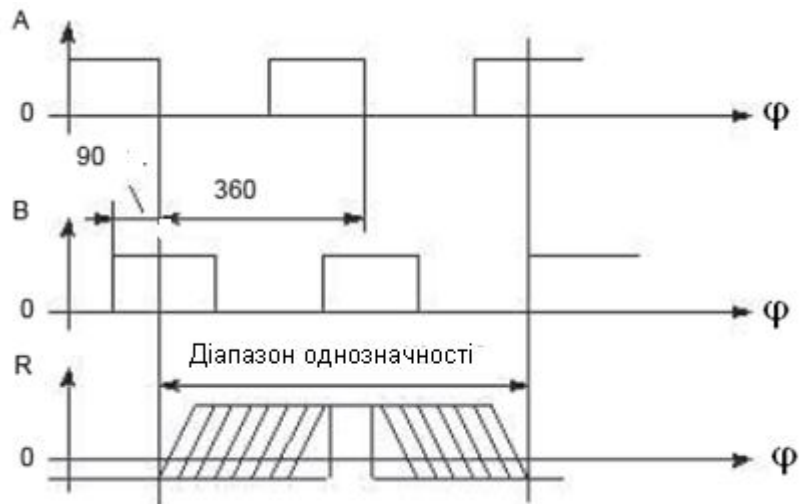


Рисунок 5.7 – Діаграми сигналів TTL інкрементального датчика

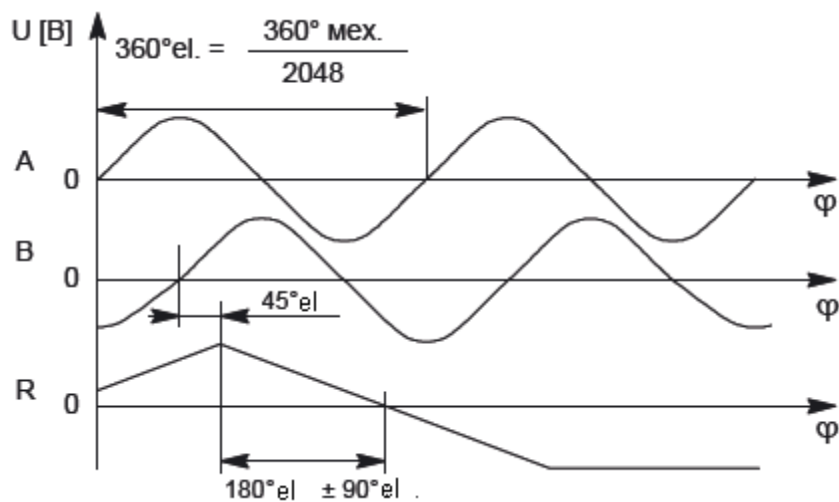


Рисунок 5.8 – Діаграми сигналів sin/cos рівня напруги $1 V_{pp}$ інкрементального датчика

У кодових датчиках інформація про положення створюється за допомогою N доріжок, кожна з яких відображає 1 біт інформації (рис. 5.9).

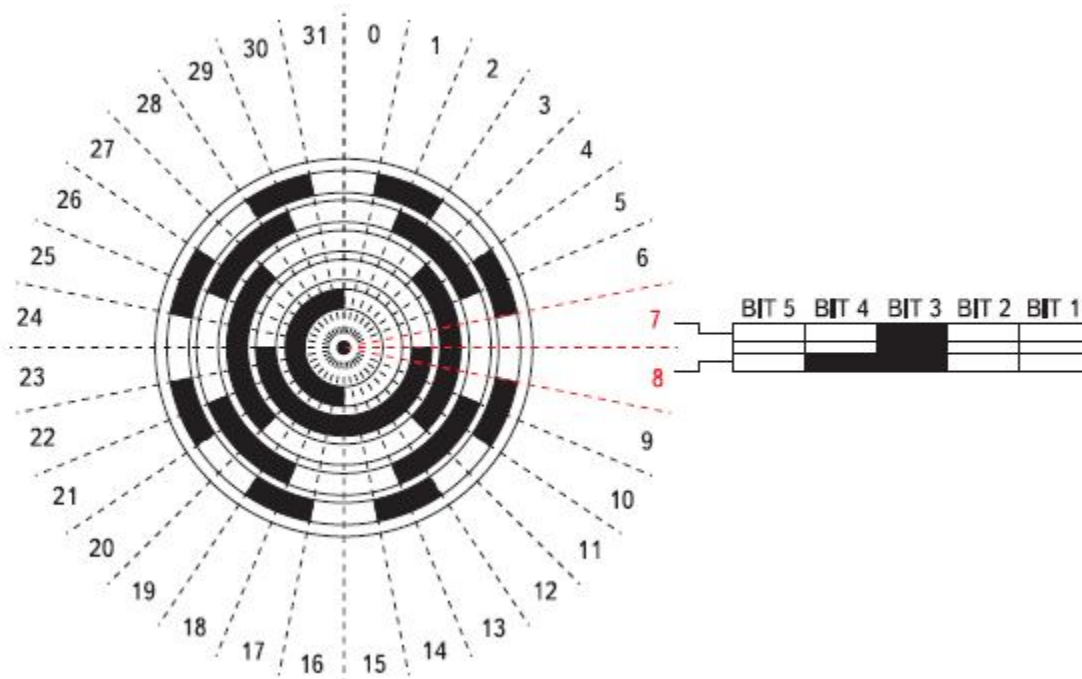


Рисунок 5.9 – Принцип створення коду положення в абсолютному датчику

Розмір двоїчного коду N залежить від конструкції датчика. Однооборотні датчики послідовного інтерфейсу SSI мають дозвіл 12 біт (4096 дискрет на оберт), а багатооборотні – 24 біта (4096 дискрет на оберт перемножуються на 4096 обертів). Такі ж характеристики мають датчики з інтерфейсом PROFIBUS-DP. У датчиків з кодером інтерфейсу по даним (Endat) дозвіл на 1 оберт в 2 рази більше – 8192 дискрет, кількість обертів – 4096.

Слід урахувати, що інкрементальні й кодові датчики є оптоелектричними приладами й живляться напругою постійного струму.

Резольвери на відміну від енкодерів являють собою електричну машину, яка живиться від спеціального джерела синусоїдальної напруги.

Резольвер – це синусно-косинусний обертовий трансформатор (СКВТ) із чотирма обмотками – двома на статорі й двома на роторі (обмотки: збудження, квадратурна, синусна й косинусна). Величина повороту вала двигуна визначається по зрушенню синусної й косинусної складових напруги.

Резольвер генерує послідовність синусоїдальних і косинусоїдальних хвильових імпульсів аналогової напруги, які визначають абсолютне положення вала в межах одного оберту. Ці аналогові напруги зазвичай перетворюються в цифрові сигнали інтерфейсною платою резольвера.

Резольвери відрізняються від енкодерів не тільки технічним виконанням, але й монтажем. В енкодері система електронного керування, як правило, вбудована в корпус, що мінімізує роботи з підключення, але обмежує діапазон робочих температур. Напроти, при створенні системи керування з резольвером, джерело живлення резольвера й інтерфейсна плата перебувають по-

за зоною робочих температур. Це дозволяє експлуатувати резольвер у середовищі з підвищеною температурою (до 100°C).

Практично всі сучасні приводи головного руху (змінного й постійного струму) допускають використання енкодерів у лінії зворотного зв'язку й лише для невеликої частини з них необхідно застосовувати резольвери.

У приводах подачі з робочими органами, що рухаються прямолінійно, для реєстрації положення застосовуються, головним чином, лінійні вимірювальні системи.

На верстатах із ЧПУ застосовуються закриті (захищені від забруднень) вимірювальні системи. Закриті датчики лінійних переміщень поставляються у двох виконаннях:

- з великим профілем корпуса (вимірювана довжина до 30 м);
- з дрібним профілем корпуса (вимірювана довжина до 1240 мм).

У корпусі закріплена склокерамічна шкала, на якій нанесені тонким шаром хрому розподільні штрихи (технологія DIADUR), причому точність розподілів лежить у межах мікрометрів.

На рисунку 5.10 показана конструкція датчика лінійних переміщень.

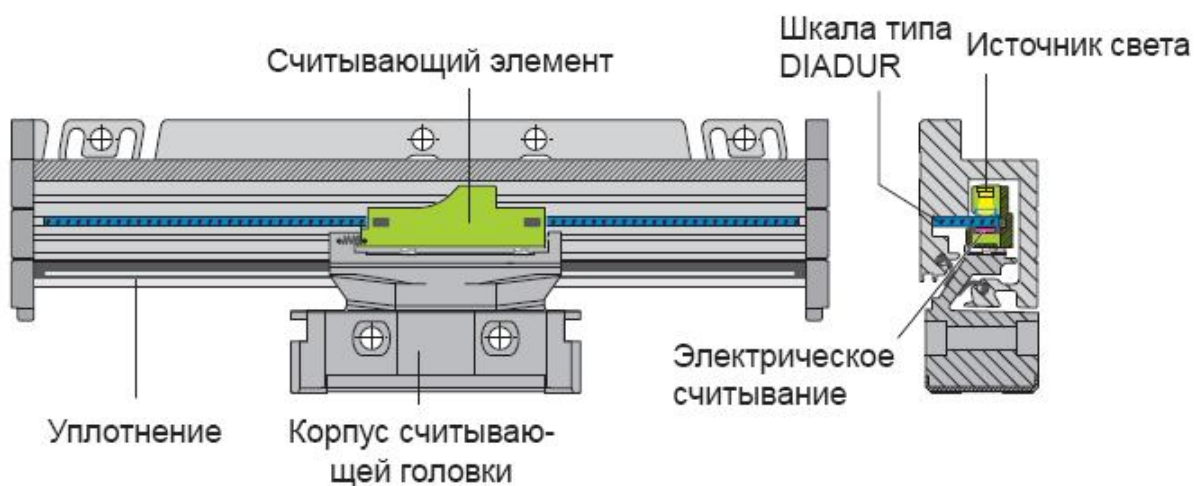


Рисунок 5.10 – Конструкція датчика лінійних переміщень фірми HEIDENHEIN

5.3 Огляд приводних модулів SIMODRIVE

Приводний модуль є, по суті, частотним перетворювачем. Він містить у собі силовий модуль (мостову схему із шістьма IGBT транзисторами) і плату керування, яка формує керуючі імпульси для силового модуля. Плата керування вставляється в силовий модуль, як показано на рисунку 5.11.

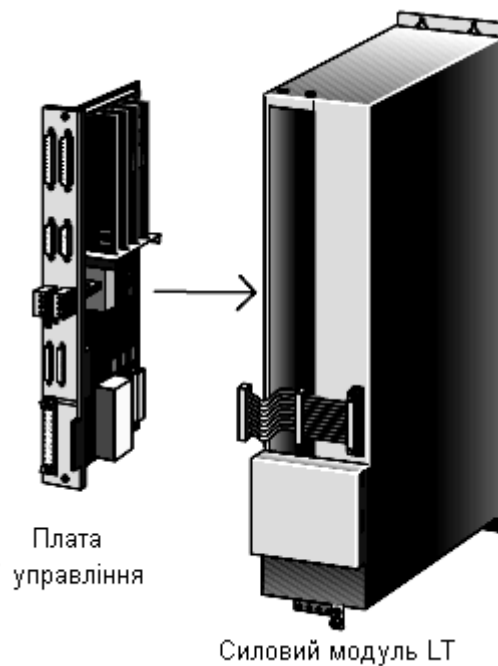


Рисунок 5.11 – Компоновка приводного модуля

Силові модулі різняться по виду охолодження в шафі (можливе виконання для природнього або примусового охолодження), по кількості осей, що обслуговуються (можливо 1-осьове й 2-осьове виконання силового модуля) і по величині номінального струму (діапазон від 3 до 200А). Залежно від величини струму в навантаженні ширина силового модуля може бути рівна 50, 100, 150 або 300 мм. Усі модулі мають однакову висоту 480 мм. Вхідна напруга постійного струму у всіх силових модулів повинна бути 600-680 В, при цьому вихідна напруга трифазного струму на двигуні досягає 430 В.

Плати керування вибираються згідно із технологіями використання приводів. Під цим розуміється тип інтерфейсу заданого значення, типи двигунів, що підключаються, типи датчиків, а також виконання (1-осьове або 2-осьове).

Плати керування діляться на наступні типи:

- плати з аналоговим ± 10 В інтерфейсом для заданого значення положення й швидкості (приводний модуль SIMODRIVE 611A для системи ЧПУ SINUMERIK 802C);
- плати із цифровим інтерфейсом заданого значення через приводну шину (приводний модуль SIMODRIVE 611D для систем ЧПУ SINUMERIK 810D і SINUMERIK 840D);
- плати універсальні (приводний модуль SIMODRIVE 611U для систем ЧПУ SINUMERIK 802C/802D і SINUMERIK 840Di), у яких є аналоговий інтерфейс ± 10 В і опціонально передбачена комунікація PROFIBUS-DP – цифровий послідовний інтерфейс.

Той або інший тип інтерфейсу плати керування вибирається виходячи з вимог до умов роботи привода, типу двигуна, а також установлених на двигуні датчиків.

Плати з аналоговим інтерфейсом заданого значення

Плати з аналоговим керуванням можна застосовувати для приводів із синхронними двигунами 1FT5, 1FK6 і асинхронними типу 1PH на автоматичних лініях, маніпуляторах, простих верстатах або верстатах зі спільними завданнями позиціонування *без високих вимог до якості регулювання й точності позиціонування*. Інформація про фактичне значення числа обертів, позиції ротора двигуна й фактичним значенні положення надходить від вбудованого у двигун датчика – резольвера.

Для роботи із трифазними серводвигунами типу 1FT5 поставляються два варіанти плат з однаковою якістю регулювання, але з різними інтерфейсами передачі сигналів до системи ЧПУ верстата на рівень оператора – інтерфейсом Standard і інтерфейсом Komfort. Плати керування з інтерфейсом Standard є в одно- і двохосьовім виконанні. Плати керування з інтерфейсом Komfort є тільки в одноосьовім виконанні.

При роботі із трифазними асинхронними двигунами 1PH плата керування з інтерфейсом Komfort дозволяє реалізувати функції керування приводом головного руху при наїзді на жорсткий упор, при обмеженні заданого значення струму й ін. Плата має входи для підключення кінцевих вимикачів BERO (рис. 5.12), а також інкрементального датчика двигуна із сигналами sin/cos 1Vpp або датчика SIMAG H.

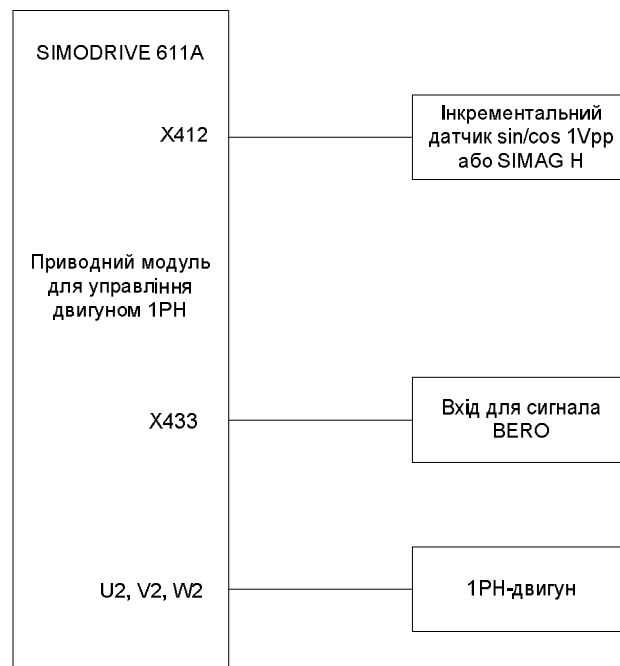


Рисунок 5.12 – Підключення приводного модуля із платою керування SIMODRIVE 611A (аналоговий інтерфейс)

Максимальна частота струму двигуна становить 1100 Гц.

У частотному діапазоні двигуна вище 10 Гц використовується алгоритм регулювання, орієнтований на поле. Завдяки цьому досягаються високі динамічні властивості й стійкість проти перекидання.

Плати з універсальним інтерфейсом заданого значення

Плати SIMODRIVE 611 universal HR – це плати керування з аналоговим інтерфейсом заданого значення числа обертів і опціональним інтерфейсом PROFIBUS DP, а також як варіант із функціональністю позиціонування.

Плати SIMODRIVE 611U використовуються для роботи із синхронними двигунами 1FT6/1FK/1FN/1FW6/1FE1, асинхронними двигунами 1PH, лінійними двигунами 1LA, а також із двигунами інших виробників, якщо вони підходять для роботи з перетворювачем.

Зовнішній вигляд плати SIMODRIVE 611 universal показано на рисунку 5.13. До цих плат можуть бути підключені наступні типи датчиків:

- 1) резольвер з дозволом 14 або 12 біт, кількістю пар полюсів від 1 до 6, робочою частотою зчитування f_g макс. до 108/432 Гц і внутрішнім множенням імпульсів до 4096/16348;



Рисунок 5.13 – Зовнішній вигляд плати SIMODRIVE 611 universal

- 2) інкрементальний датчик із сигналами \sin/\cos 1V_{pp}, дозволом до 65535 імпульсів із внутрішнім множенням, із частотою зчитування до 350 кГц;
- 3) абсолютний датчик з інтерфейсом Endat;
- 4) датчик TTL (інтерфейс RS422) для асинхронних двигунів із частотою зчитування імпульсів до 420 кГц.

Слід урахувати, що до плат з 1-осьовим виконанням можуть

підключатися тільки резольвери, а до плат з 2-осьовим виконанням – зовнішні вимірювальні системи (абсолютні й інкрементальні), а також датчик, установлений на двигуні.

Плати SIMODRIVE 611U використовуються для керування приводами на верстатах нормальної точності.

Плати керування із цифровим інтерфейсом заданого значення

Плати керування SIMODRIVE 611D із цифровим інтерфейсом заданого значення для VSA і HAS використовуються в комбінації з наступними двигунами:

- трифазними серводвигунами 1FT6/1FK6 і лінійними двигунами 1FN для приводів подачі;
- трифазними двигунами 1PH-/1FE1 для приводів головного руху;
- вбудованими тороїдальними електродвигунами 1FW6 для прямих приводів з високою віддачею моменту обертання.

Плати керування обробляють вбудований у двигун інкрементальний датчик sin/cos $1V_{pp}$ з дозволом до 4,2 млн. інкрементів/оберт двигуна або кодовий (абсолютний) датчик.

До плат керування може бути підключена пряма вимірювальна система (DMS) із застосуванням інкрементальних датчиків із синусно-косинусними сигналами (рис. 5.14).

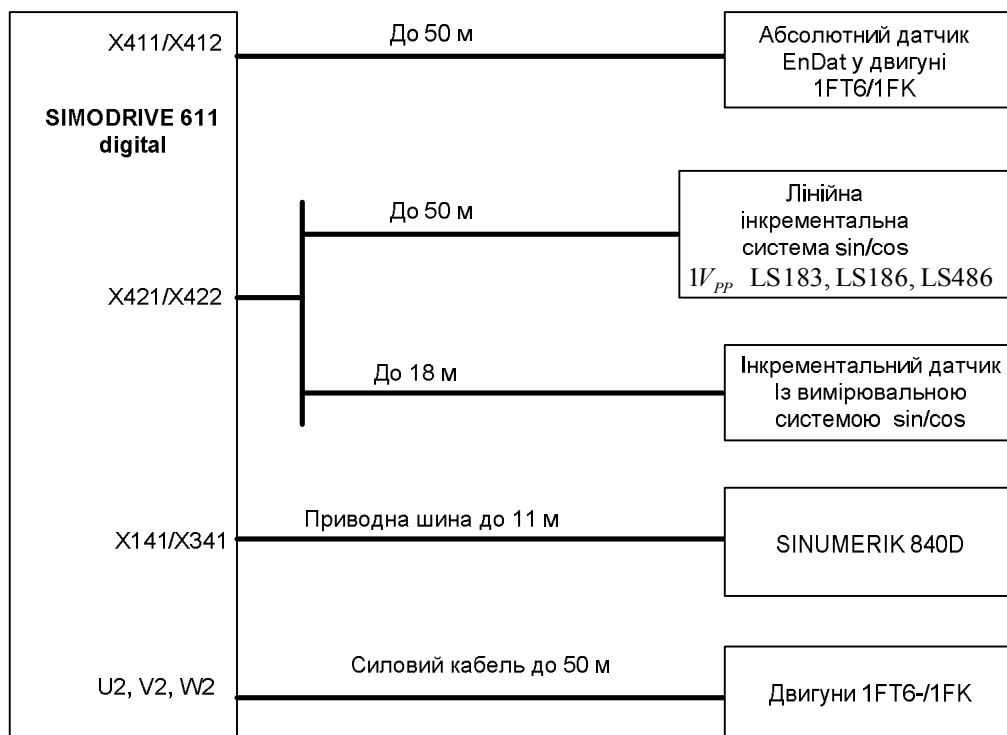


Рисунок 5.14 – Підключення плати керування із цифровим інтерфейсом заданого значення (привод подачі)

Сигнали з датчиків положення можуть оброблятися в пристрої ЧПУ SINUMERIK. Їхнє введення в ЧПУ здійснюється через цифрову приводну шину.

Плати керування із цифровим інтерфейсом заданого значення можуть використовуватися універсально в приводі подачі або головного руху.

Програмне забезпечення для плат управління вбудовано в системах SINUMERIK 810D/840D/840C. При кожнім включенні СЧПУ й приводів програмне забезпечення завантажується в цифрові плати керування. При введенні в експлуатацію через конфігурацію привода визначається, чи йде мова про привод подачі або головного руху.

У плат керування із цифровим інтерфейсом заданого значення можна вибирати між керуванням Standard і більш продуктивним керуванням Performance.

Для в SINUMERIK 840D плати керування доповнені серіями "High Standard" і "High Performance". Обидва варіанта використовують ідентичні інтерфейси приводів і однакове програмне забезпечення для регулювання.

Відмінні риси при керуванні High Standard і High Performance полягають у наступному:

- більші обчислювальні можливості й більший обсяг пам'яті;
- наявність 1 або 2 входу для підключення датчиків двигуна й для введення напруги від прямої вимірювальної системи;
- наявність входів для підключення датчиків BERO;
- підтримка Safety Integrated (інтегрована безпека) апаратними засобами.

Деякі характеристики плат керування із цифровим інтерфейсом наведено в таблиці 5.1.

Цифрова плата керування SIMODRIVE 611 HLA для керування гідравлічним лінійним приводом (HLA)

Цифрова плата керування SIMODRIVE 611 HLA призначена для керування електрогідравлічними регульовальними клапанами гідравлічних лінійних осей у комбінації з SINUMERIK 840D. Плата може обслуговувати дві гідравлічні осі. Після вставки в універсальний порожній корпус шириною 50 мм виходить модуль HLA.

Плата постачена електричними інтерфейсами (приладова шина, приводна шина й шина проміжного контуру), що дозволяє інтегрувати її в структуру приводів SIMODRIVE 611.

Плата керування HLA створює силове живлення для регульовальних і запірних клапанів. Можливий і змішаний режим однієї осі HLA і однієї аналогової осі ANA з інтерфейсом заданого значення числа обертів ± 10 В.

Таблиця 5.1

Дані	Плата керування			
	Standard	High Standard	Performance	High Performance
Макс. частота вихідної напруги	600 Гц	600 Гц	1200 Гц	1400 Гц
Гранична частота датчика двигуна	200 кГц	200 кГц	300 кГц	350 кГц
Множення імпульсів:	128	128	2048	2048
Макс. довжина кабелю датчика із сигналом напруги	50 м	50 м	50 м	50 м
Датчик двигуна й прямі вимірювальні системи (DMS)				
Інкрементальний датчик sin/cos 1 Vpp	так	так	так	так
Абсолютний датчик Endat	так	так	так	так
Safety Integrated із внутрішньою заборною імпульсів через приводну шину	немає	так, при керуванні з DMS	немає	так, при управл. з DMS
Краща сфера використання	Стандартні серійні верстати		Верстати для точної обробки	

5.4 Модулі живлення

Лінійка приводів SIMODRIVE 611 розрахована на роботу від трифазної мережі (3 AC) із заземленою нейтраллю (TN).

Структура привода підключається до цієї мережі напругою 400 В, 415 В або 480 В через модуль живлення (NE). Якщо мережа має іншу напругу, наприклад 220 В, для адаптації модуля живлення до мережі необхідно застосувати відповідний трансформатор.

Модулі живлення підрозділяються на регульовані (E/R), які забезпечують не тільки живлення, але й рекуперацію, і нерегульовані (UE). Модулі живлення створюють постійну напругу для проміжного контуру, а також напруги для живлення електроніки.

Напруга на проміжному контурі (вихідна напруга модуля живлення) залежить від типу модуля. Нерегульовані модулі UE створюють вихідну напругу *постійного струму* від 490 до 680В, Регульовані модулі E/R можуть створювати напруги 600/625/680 В.

Напруги для живлення електроніки ($\pm 24\text{ В}$, $\pm 15\text{ В}$, $+5\text{ В}$) подаються від модуля живлення через приладову шину на приводні модулі й, при необхідності, на модулі SINUMERIK 840D або SINUMERIK 810D.

Модуль живлення постачаний розніманнями для введення сигналів готовності, блокувань і захистів, які забезпечують безпечну роботу модуля й підключених до нього приводів. У структурі привода модуль живлення розташовується першим ліворуч (рис. 5.15). Далі встановлюються силові модулі. Ширина модуля живлення залежить від його потужності – 50, 100, 200 і 300 мм для потужності від 5 кВт до 120 кВт. Уніфікована висота всіх модулів становить 480 мм.

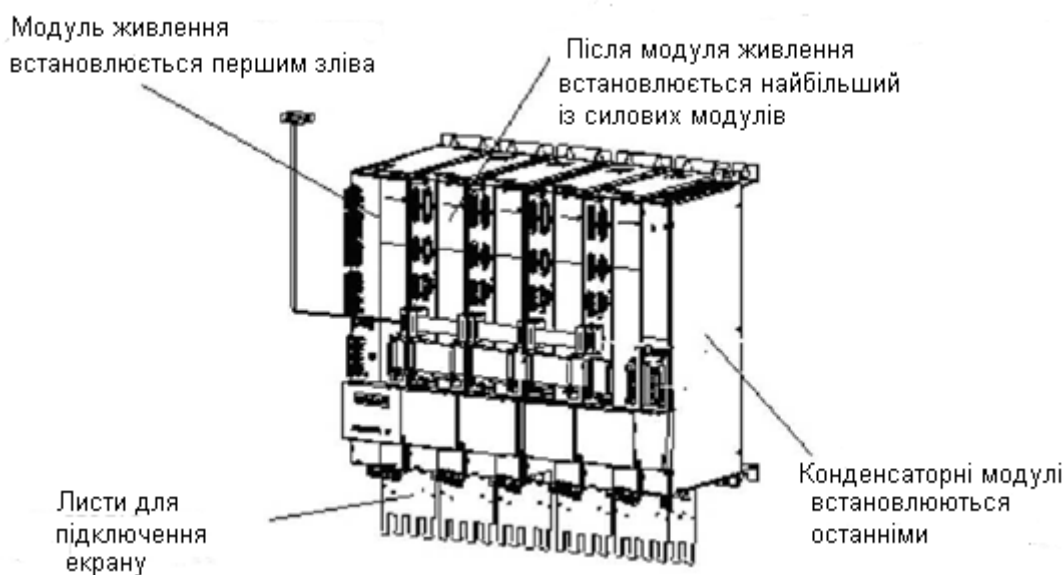


Рисунок 5.15 – Розташування модулів у структурі приводів

Охолодження модулів здійснюється або розсіюванням тепла усередині шафи (природне охолодження), або вентиляторами (примусове охолодження).

Для обмеження впливу перетворювача на мережу між мережею й модулем живлення доцільно включити мережний фільтр і комутуючий дросель HF. Комутаційні дроселі HFD і імпульсний опір 1,5/25 кВт необхідні, якщо в структурі приводів перебувають лінійні приводи, тороїдальні електродвигуни, двигуни інших виробників або можливі явища резонансу.

У модулях E/R з боку проміжного контуру можна включити конденсаторний модуль, який дозволяє нагромадити енергію гальмування, а також створити перекриття при короткочаснім відключенні живлення.

У нерегульованих модулях живлення UE у ланцюг проміжного контуру потрібно включити модуль імпульсного опору, який призначений для відводу надлишкової енергії, що виникає при гальмуванні.

Схема подачі живлення на перетворювач із контролем запобіжників наведено на рисунку 5.16.

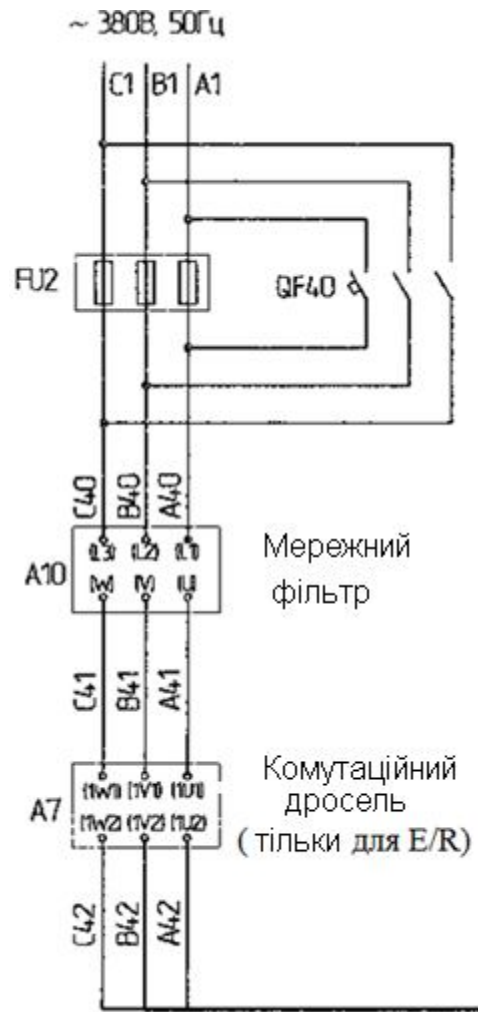


Рисунок 5.16 – Схема подачі живлення на перетворювач

6 ПРОЕКТУВАННЯ СТРУКТУРИ ПРИВОДА SIMODRIVE

Проектування структури приводів SIMODRIVE здійснюється у два етапи:

Етап 1 – вибір компонентів.

Етап 2 – конфігурація з'єднань.

На першому етапі необхідно вибрати:

1. Двигун, датчики й плати управління.
2. Силові модулі.
3. Модулі живлення.

Після вибору всіх компонентів привода необхідно перевірити й урахувати всі умови по підключенню до мережі, забезпечити електромагнітну сумісність компонентів, скласти схеми з'єднання апаратури й розробити компонування шафи.

6.1 Принципи вибору двигунів, датчиків і плат керування

Вибір двигуна повинен здійснюватися *на основі механічних і динамічних вимог*, пропонованих до двигуна. Вимоги до перевантажувальної здатності двигуна залежать від висоти й кількості пікових навантажень під час експлуатації.

Для захисту двигунів необхідно використовувати захисні вимикачі. При перевантаженні двигуна повинне бути передбачене включення сигнального контакту.

Також необхідно передбачити контроль з'єднання двигуна із силовим модулем. Якщо двигун при експлуатації з активованим дозволом імпульсів від'єднається від силового модуля, то існує небезпека саморуйнування силової частини й плати керування.

Якщо до двигуна приєднане стояночне гальмо, то воно повинне приводитися тільки в стані спокою двигуна. Слід звернути особливу увагу на використання стояночного гальма при висячих вантажах, тому що тут є високий потенціал небезпеки.

Певному типу двигуна відповідає тип вбудованого датчика. Підпорядкування датчика й плати управління певному типу двигуна показано на рисунку 6.1.

Плата управления резольвера							Тип двигателя		Датчик
Плата управления подачи с интерфейсом Standard и Komfort									
Плата управления главного движения с аналоговым интерфейсом заданного значения									
Плата управления привода Digital Performance 1 или High Performance (режим VSA)									
Плата управления привода Digital Performance 1 или High Performance (режим HSA)									
Плата управления привода Standard 2 или High Standard Digital									
Плата управления привода 611 universal HR – резольвер									
Плата управления привода 611 universal HR – Vpp сигналы напряжения									
д	а						1FT5 серводвигатель	Трехфазный тахометр и RLG	
д	а						1FT5 серводвигатель	Трехфазный тахометр и RLG с опциональным встроенным/встроенным инкрементальным или абсолютным датчиком	
д	а					д	1FK6 серводвигатель	Резольвер	
			д	а		д	1FT6 / 1FK6 серводвигатель	Инкрементальный датчик 1 Vpp	
			д	а		д	1FT6 / 1FK6 серводвигатель	Абсолютный датчик Multiturn	
		д	а			д	1PH4 / 6 / 7 двигатель главного движения	Инкрементальный датчик 1 Vpp	
		д	а			д	1FE1 / 1PH2 двигатель главного движения	Инкрементальный датчик (датчик полого вала)	
		да	да	да	да	да	Стандартный двигатель	без датчика	

Рисунок 6.1 – Підпорядкування датчика й плати керування певному типу двигуна

При проектуванні системи привода необхідно зробити наступне.

По-перше, відразу потрібно визначитися з організацією вимірювальної системи – прийняти непряму або пряму реєстрацію положення робочого органа привода. Непряма реєстрація положення передбачає використання датчика, встановленого на двигуні, а пряма реєстрація будується на використанні лінійних або кругових датчиків, встановлюваних безпосередньо на робочому органі. У приводах головного руху робочим органом є шпindel, а в приводах подачі робочим органом є вузол переміщення деталі (стіл) або переміщення інструмента (супорт із різцетримачем). Слід урахувати, що пряма вимірювальна система забезпечує більшу точність реєстрації положення робочого органа, тому що в слідкуючій системі вона охоплює зворотним зв'язком усі елементи привода. Однак така система вимагає установки додаткових механізмів і пристроїв.

По-друге, при проектуванні системи привода необхідно розв'язати питання про інтерфейс заданого значення швидкості й положення – вибрати аналогове або цифрове управління.

Слід ураховувати, що аналогове управління застосовується для приводів без високих вимог до якості регулювання, точності позиціонування й ди-

наміки.

Розглянемо варіанти непрямой й прямої реєстрації швидкості й положення з аналоговим і цифровим інтерфейсами заданого значення.

6.2 Непряма реєстрація положення з аналоговим і цифровим інтерфейсами

На рисунку 6.2 показаний варіант управління з аналоговим інтерфейсом заданого значення й позиціонування *привода подачі* з керуванням від ЧПУ. Для позиціонування використовується інтерфейс *копії імпульсного датчика WSG* – положення ротора, яке реєструє револьвер, представляється на ЧПУ в вигляді сигналів імпульсного датчика. Імітація датчика двигуна (резольвера) здійснюється через диференціальні квадратурні синусні (А) і косинусні (В) сигнали згідно RS 485, RS 422 (TTL), а також через сигнал референтної мітки початку оберту R. Резольвер дозволяє одержати на один оберт $1024 \times 4 = 4096$ імпульсів (із учетверінням), що відповідає точності датчика 0,12 град.

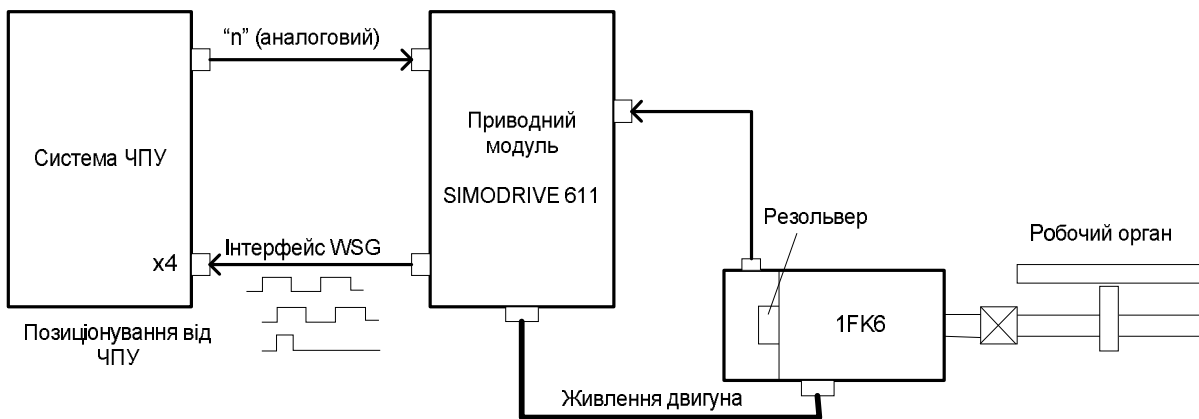


Рисунок 6.2 – Керування приводом подачі із застосуванням інтерфейсу WSG

На рисунку 6.3 показаний варіант керування приводом *головного руху* із застосуванням інкрементального датчика. Точність такого датчика досягає 0,006 град. завдяки тому, що число імпульсів (кроків) на оберт вала може бути значно примножено. Розрахунки числа кроків на один оберт проводяться по формулі:

$$M = 4k \times Z,$$

де k – установлюваний у приводі коефіцієнт множення (0,5; 1; 2; 4), Z – число штрихів на оптичному диску датчика.

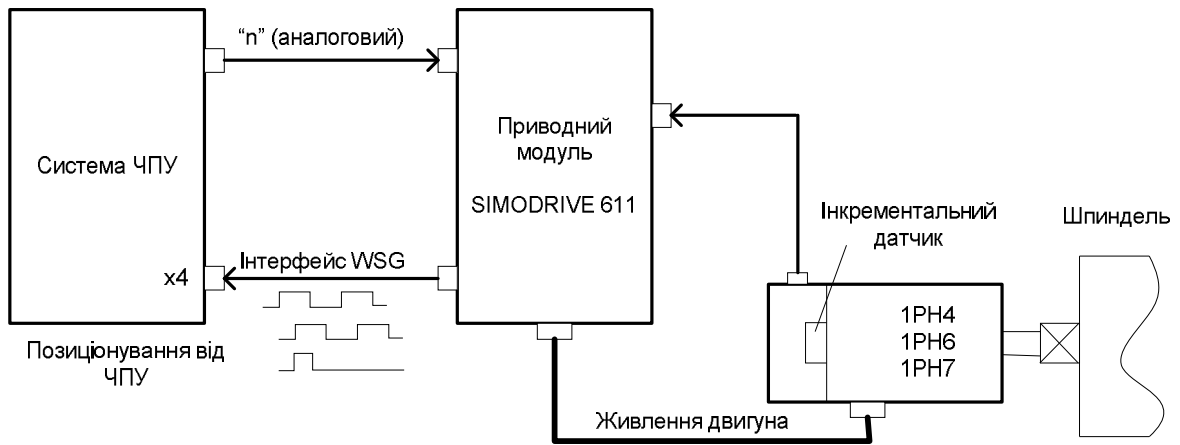


Рисунок 6.3 – Управління приводом головного руху з інтерфейсом WSG

Так, наприклад, для $Z = 2048$ штрихів при застосуванні множення на 4 одержимо:

$$M = 4 \times 4 \times 2048 = 32768 \text{ кроків/оберт.}$$

Розділивши 360° на отримане число кроків, одержимо ціну одного кроку, тобто погрішність виміру кута повороту:

$$G_{расч} = \frac{360}{32768} = 0,011 \text{ град.}$$

Для позиціонування привода за допомогою ЧПУ можливий також варіант із використанням опціонального датчика.

Нехай, наприклад, потрібно створити систему управління приводом подачі із двигуном 1FT5. Для цього двигуна передбачена плата управління з інтерфейсом Standard і інтерфейсом Komfort. Для керування двигуном у каналі зворотного зв'язку передбачені тахогенератор і датчик кута повороту ротора RLG. Позиціонування здійснюється за допомогою опціонального інкрементального або абсолютного датчика, приєднаного до ходового гвинта й підключеного до пристрою ЧПУ. Керування швидкістю й положенням вала двигуна здійснюється від ЧПУ аналоговим сигналом (± 10 В). У цьому варіанті керування точність позиціонування залежить від точності опціонального датчика й методу обробки сигналу в ЧПУ.

Конфігурація такого керування наведена на рисунку 6.4.

Непряма реєстрація положення й швидкості переміщення робочого органу *при цифровім керуванні* приводами подачі і головних рухів здійснюється принципово однаково.

Базове виконання системи керування показано на рисунку 6.5.

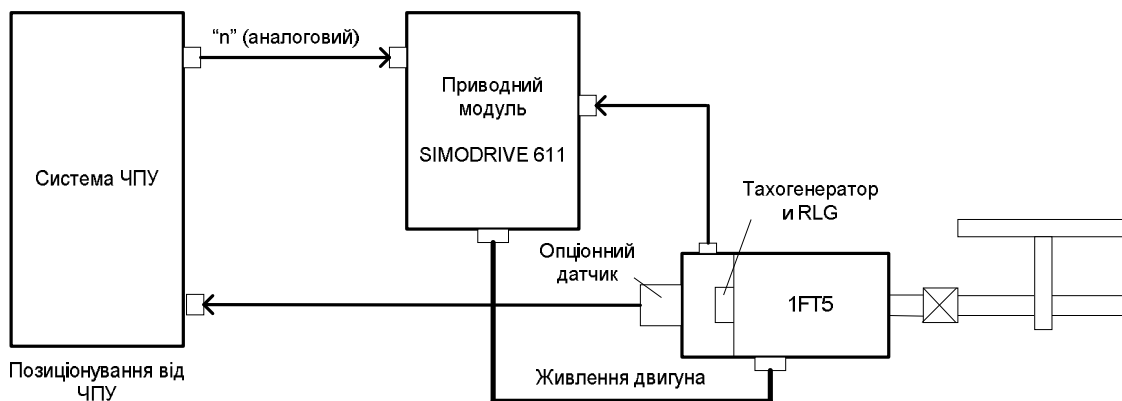


Рисунок 6.4 – Керування приводом подачі з використанням опціонального датчика

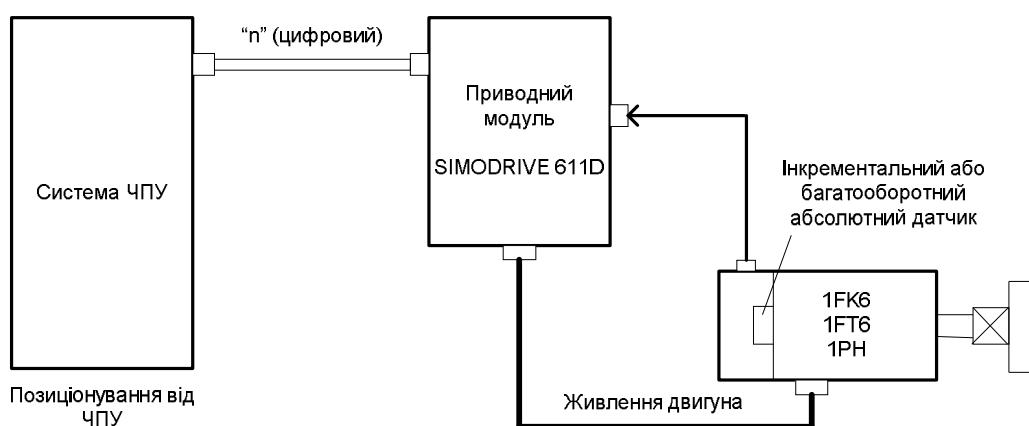


Рисунок 6.5 – Непряма реєстрація положення й швидкості із цифровим інтерфейсом

6.3 Пряма реєстрація положення з аналоговим керуванням

Пряма реєстрація положення забезпечує більш високу точність керування й позиціонування, чим непряма реєстрація, тому що виключає впливи таких ефектів, як неточності виготовлення гвинта, а також люфти й пружні деформації в передачі.

При керуванні *приводом подачі* для прямої реєстрації положення доцільно застосовувати лінійні вимірювальні системи.

На рисунку 6.6 показаний варіант побудови привода із двигуном 1FK6 і аналоговим інтерфейсом заданого значення Standard.

У якості датчика зворотному зв'язку для забезпечення векторного керування двигуном застосовується резольвер, а для забезпечення зворотного зв'язку *по положенню* використовується лінійний датчик, сигнал з якого подається на систему ЧПУ. Точність позиціонування залежить від точності датчика й методу обробки в ЧПУ. Так, наприклад, для датчика типу LS 187 фірми HEIDENHAIN точність становить 5 мкм.

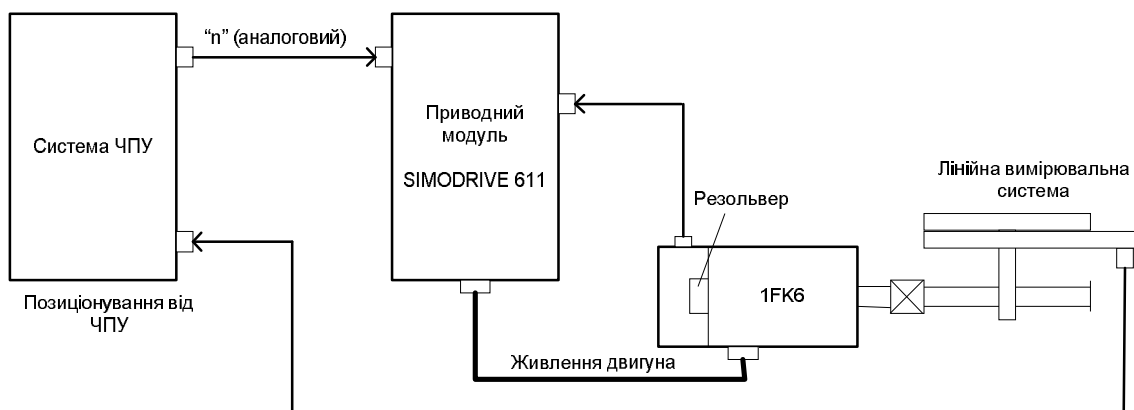


Рисунок 6.6 – Конфігурація привода із прямою реєстрацією положення лінійною вимірною системою

Якщо в приводі подачі застосований двигун 1FT5, то для керування цим двигуном замість резольвера застосовується тахогенератор (для контролю швидкості) і датчик положення ротора RLG (для управління перетворювачем).

При керуванні приводом *головного руху* є кілька варіантів прямої реєстрації положення й позиціонування шпинделя.

Якщо не потрібна висока точність, то позиціонування можна здійснювати з використанням датчиків BERO. Для підключення цих датчиків слід передбачити модуль KLEMMEN, який вставляється в гніздо плати керування приводного модуля SIMODRIVE 611 universal. Модуль KLEMMEN дозволяє вільно параметризувати функціональність восьми входів і восьми виходів.

Конфігурація привода шпинделя з датчиком позиціонування BERO показано на рисунку 6.7.

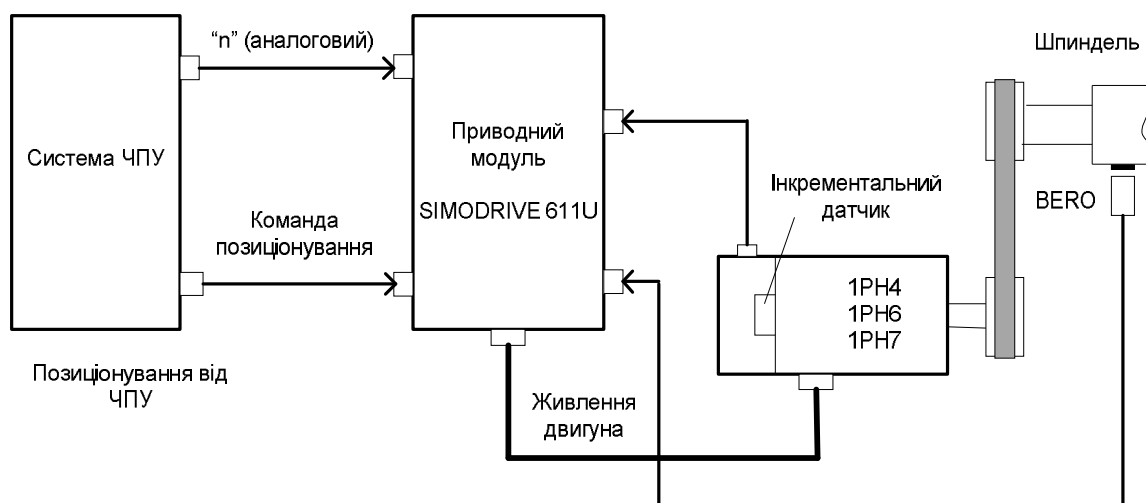


Рисунок 6.7 – Позиціонування шпинделя із датчиком BERO

Команда на позиціонування подається із ЧПУ. Слід урахувати, що точність датчика BERO залежить від часу його перемикавання, гістерезису, часу обробки сигналу в ЧПУ й швидкості обертання шпинделя.

Для керування двигуном 1РН застосовується інкрементальний датчик з кількістю штрихів на диску $Z=2048$. Точність контролю кута повороту ротора можна збільшити шляхом множення числа імпульсів датчика з максимальним коефіцієнтом 2048.

При високих вимогах до точності позиціонування шпинделя замість датчика BERO слід використовувати інкрементальний або абсолютний датчик. Варіант підключення датчика шпинделя показаний на рисунку 6.8.

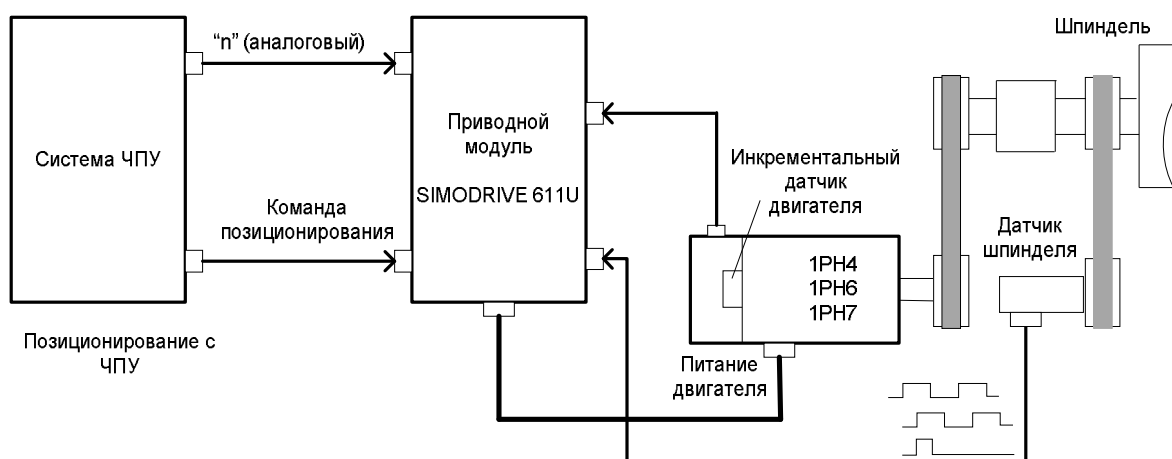


Рисунок 6.8 – Привод головного руху із прямою реєстрацією положення шпинделя

Для прямої реєстрації положення шпинделя застосований інкрементальний датчик, який з'єднаний зі шпинделем за допомогою ремінної передачі. При подачі команди позиціонування на плату керування SIMODRIVE 611U проводиться перемикавання на внутрішнє керування положенням по різниці заданого й фактичного положення.

У деяких випадках потрібно організувати керування шпинделем, як віссю С. Для розв'язку цього завдання необхідно використовувати датчик зі стандартною доріжкою й доріжкою осі С (положення шпинделя).

Графіки вихідних сигналів такого датчика наведено на рисунку 6.9. Із графіків видно, що стандартна доріжка із сигналами А і В має 2048 кроків на оберт (360° кута повороту вала). Доріжка положення шпинделя із сигналами С і D має один крок на оберт, тобто 360° електричного сигналу відповідають 360° повороту вала. При цьому референтна мітка R для обох доріжок та сама.

Слід звернути увагу на те, що на графіках масштаб осі абсцис різний.

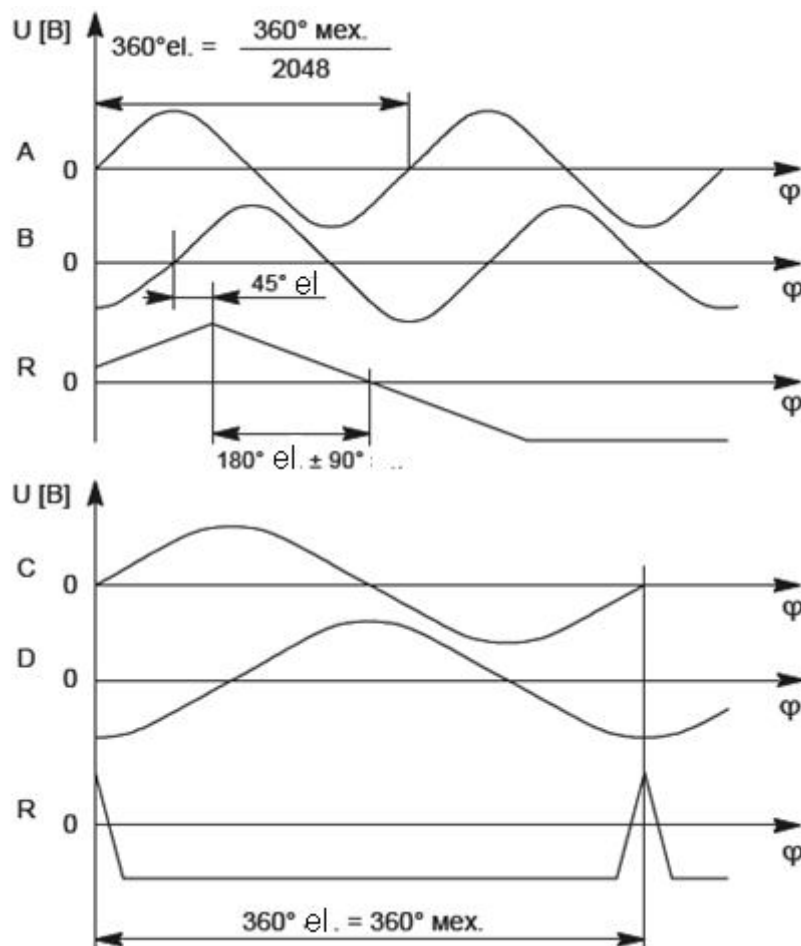


Рисунок 6.9 – Графіки сигналів інкрементального датчика, що підтримує режим осі C

На відміну від попередньої схеми (рис. 6.8) датчик шпинделя підключається не до плати керування, а до пристрою ЧПУ, наприклад, в SINUMERIK 840D до вимірювальної розетки X131, в SINUMERIK 810D – до рознімання вимірювальної системи X411.

6.4 Пряма реєстрація положення із цифровим керуванням

При використанні плати із цифровим керуванням сигнали датчиків прямої реєстрації положення робочого органа подаються на плату керування SIMODRIVE, яка отримує завдання від ЧПУ.

На рисунку 6.10 показаний варіант із використанням датчика BERO для реєстрації положення шпинделя (функція BERO для привода подачі не дозволяється).

Замість датчика BERO для прямої реєстрації положення може бути застосований круговий інкрементальний датчик з вихідними сигналами *струму* (для керування двигуном використовуються сигнали *напруги* 1Vpp). Приклад такої схеми наведено на рисунку 6.11.

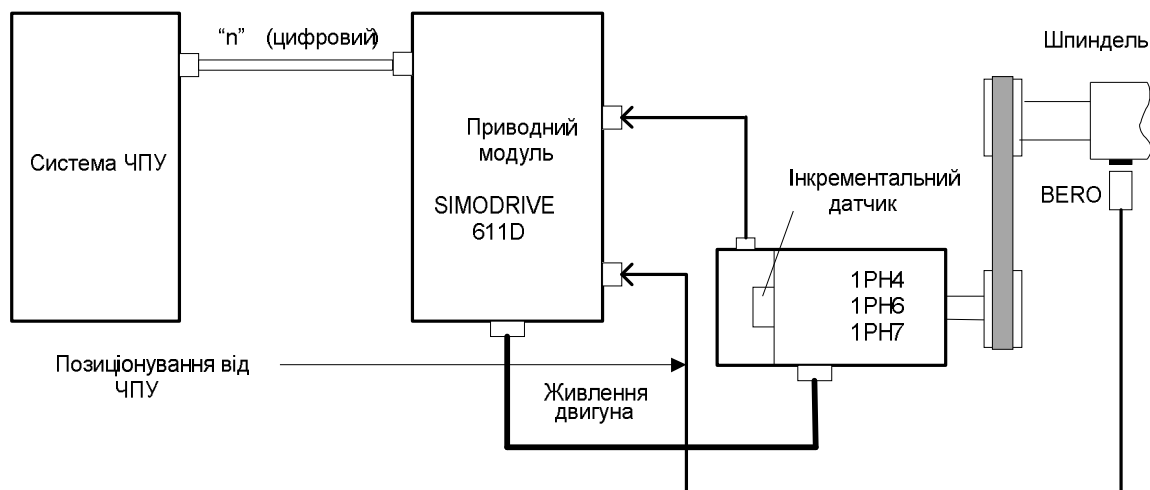


Рисунок 6.10 – Керування приводом головного руху з позиціонуванням за допомогою датчика BERO

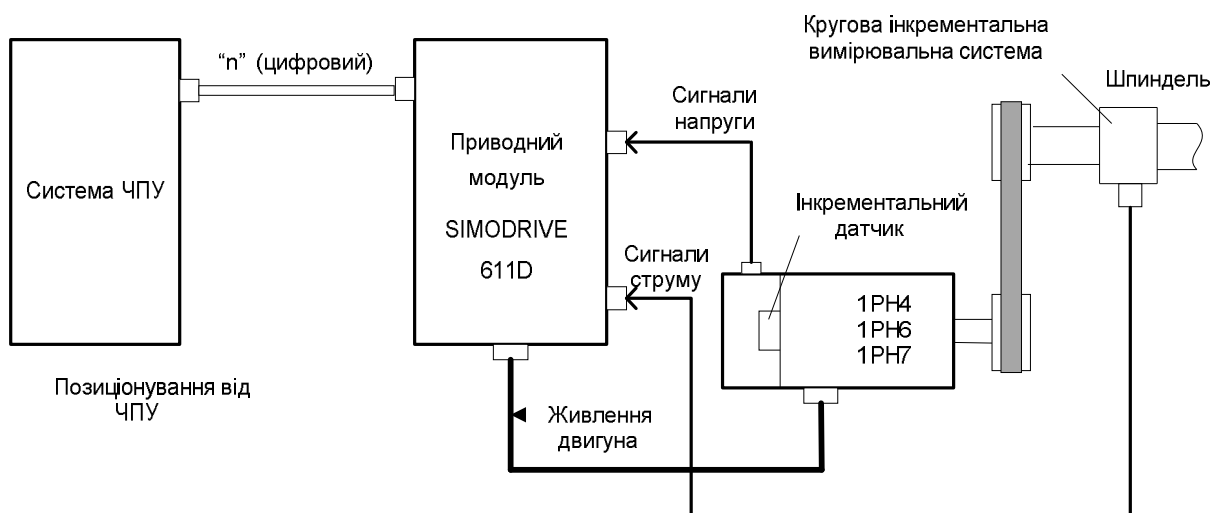


Рисунок 6.11 – Керування приводом головного руху із прямою реєстрацією положення інкрементальним датчиком

Для керування приводом подачі зазвичай застосовуються лінійні вимірювальні системи з вихідними сигналами напруги й струму.

На рисунку 6.12 показаний приклад прямої реєстрації положення робочого органа привода подачі з лінійною інкрементальною вимірювальною системою.

Плати із цифровим керуванням SIMODRIVE 611D підтримують також цифровий двунправлений інтерфейс Endat для вимірювальних систем виробництва компанії HEIDENHAIN.

На рисунку 6.13 показаний варіант підключення абсолютного датчика з інтерфейсом Endat для прямої реєстрації положення в приводі подачі.



Рисунок 6.12 – Цифрове керування приводом подачі із прямою реєстрацією положення

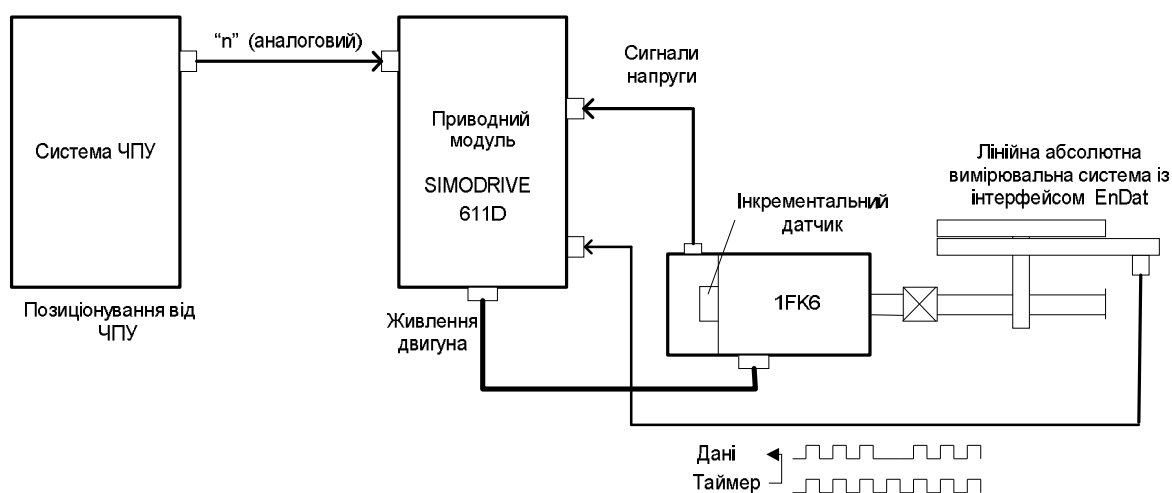


Рисунок 6.13 – Цифрове керування приводом подачі з абсолютним датчиком Endat

За допомогою інтерфейсу Endat можлива не тільки передача значень координат, але й передача інших даних, що втримуються в датчиках, їх актуалізація, зміна й збереження. Дані передаються синхронно з тактовою частотою, що задається керуючою електронікою або таймером. Тип переданих даних (значення координат, параметри, результати діагностики і т.д.) визначається командами.

В абсолютних датчиках з інтерфейсом Endat сигнали положення зчитуються із двох доріжок, а не з 13, як це проводиться в класичних конструкціях. Інкрементальні сигнали \sin/\cos обробляються й зберігаються в модулі датчика. Передача цифрових даних здійснюється по послідовному інтерфейсу з тактовою частотою 2 МГц.

6.5 Вибір і підключення модулів структури привода

Для проектування структури привода необхідно спочатку вибрати двигун (з датчиком).

Далі проводиться вибір силового модуля (LT). Основними параметрами для вибору силового модуля є: номінальний струм двигуна I_n , піковий струм I_{max} і тактова частота інвертора f_T (кГц).

При виборі силового модуля для привода подачі (VSA) потрібно керуватися даними таблиці 6.1.

Таблиця 6.1

Тип LT	Код Нр.	I_n/I_{max} [A] f_T 4,57 кГц	I_n/I_{max} [A] f_T 4,92 кГц	I_n/I_{max} [A] f_T 5,33 кГц	I_n/I_{max} [A] f_T 5,82 кГц	I_n/I_{max} [A] f_T 6,40 кГц	I_n/I_{max} [A] f_T 7,11 кГц	I_n/I_{max} [A] f_T 8,00 кГц
8 A	17	2,8/5,6	2,7/5,4	2,6/5,2	2,4/4,8	2,2/4,4	2,0/3,9	1,7/3,3
15 A	18	4,7/9,4	4,5/9,0	4,3/8,6	4,0/8,0	3,7/7,3	3,3/6,5	2,8/5,5
25 A	20	8,4/16,8	8,1/16,1	7,8/15,5	7,2/14,3	6,6/13,3	5,9/11,7	5,0/9,9
50 A	22	16/33	16/31	14/29	13/26	12/23	10/19	7/14
2x8 A	17	2x2,8/5,6	2x2,7/5,4	2x2,6/5,2	2x2,4/4,8	2x2,2/4,4	2x2,0/3,9	2x1,7/3,3
2x15 A	18	2x4,7/9,4	2x4,5/9,0	2x4,3/8,6	2x4,0/8,0	2x3,7/7,3	2x3,3/6,5	2x2,8/5,5
2x25 A	20	2x8,4/16,8	2x8,1/16,1	2x7,8/15,5	2x7,2/14,3	2x6,6/13,3	2x5,9/11,7	2x5,0/9,9
2x50 A	22	2x16/33	2x16/31	2x14/29	2x13/26	2x12/23	2x10/19	2x7/14
80 A	23	26/52	25/50	23/47	22/43	20/39	17/34	14/28
160 A	25	52/105	50/100	48/97	45/89	41/82	36/73	31/62
200 A	26	66/131	63/126	60/121	56/111	51/102	46/91	39/77
400 A	28	130/195	124/186	117/175	108/162	98/147	85/128	70/105

Слід урахувати, що передумовлена тактова частота інвертора на платі керування VSA Digital $f_T=4,0$ кГц. Із таблиці видно, що із збільшенням частоти інвертора значення струмів зменшуються.

Силовий модуль повинен бути постачений платою керування. При виборі плати слід урахувати інтерфейси завдання швидкості й положення, передачі сигналів на верхній рівень, а також можливості підключення опціональних датчиків. Для вибору плати керування слід керуватися табл. 6.2.

Таблиця 6.2

Плата управління	Вісь	Датчик двигуна	Двигун	Опціональний інтерфейс
SIMODRIVE 611U	1	Резольвер	1FT6, 1FK, 1FE1, 1FN, 1PH, 1LA	PROFIBUS DP, RS 485
SIMODRIVE 611U	2	Резольвер	1FT6, 1FK, 1FE1, 1PH, 1LA	PROFIBUS DP, RS 485
SIMODRIVE 611U	2	Інкрементальний датчик sin/cos 1Vpp Абсолютний датчик	1FT6, 1FK, 1FE1, 1PH, 1LA	PROFIBUS DP, RS 485
SIMODRIVE 611A із аналоговим інтерфейсом	2/1	Тахогенератор і датчик RLG	1FT5	
SIMODRIVE 611A із аналоговим інтерфейсом	1	Двополюсний резольвер	1FK, 1FT6	
SIMODRIVE 611A із аналоговим інтерфейсом		Інкрементальний датчик SIZAG H	1PH	
SIMODRIVE 611D із цифровим інтерфейсом	1	Інкрементальний датчик sin/cos 1Vpp, Endat, SSI	1FT6, 1FK, 1FE1, 1FN3, 1PH,	
SIMODRIVE 611D 611HLA із цифровим інтерфейсом	2	Інкрементальний датчик sin/cos 1Vpp, Endat, SSI	Гідравлічні лінійні осі	

На рисунку 6.14 показана схема підключень приводної системи SINUMERIK 840D для фрезерного верстата із ЧПУ.

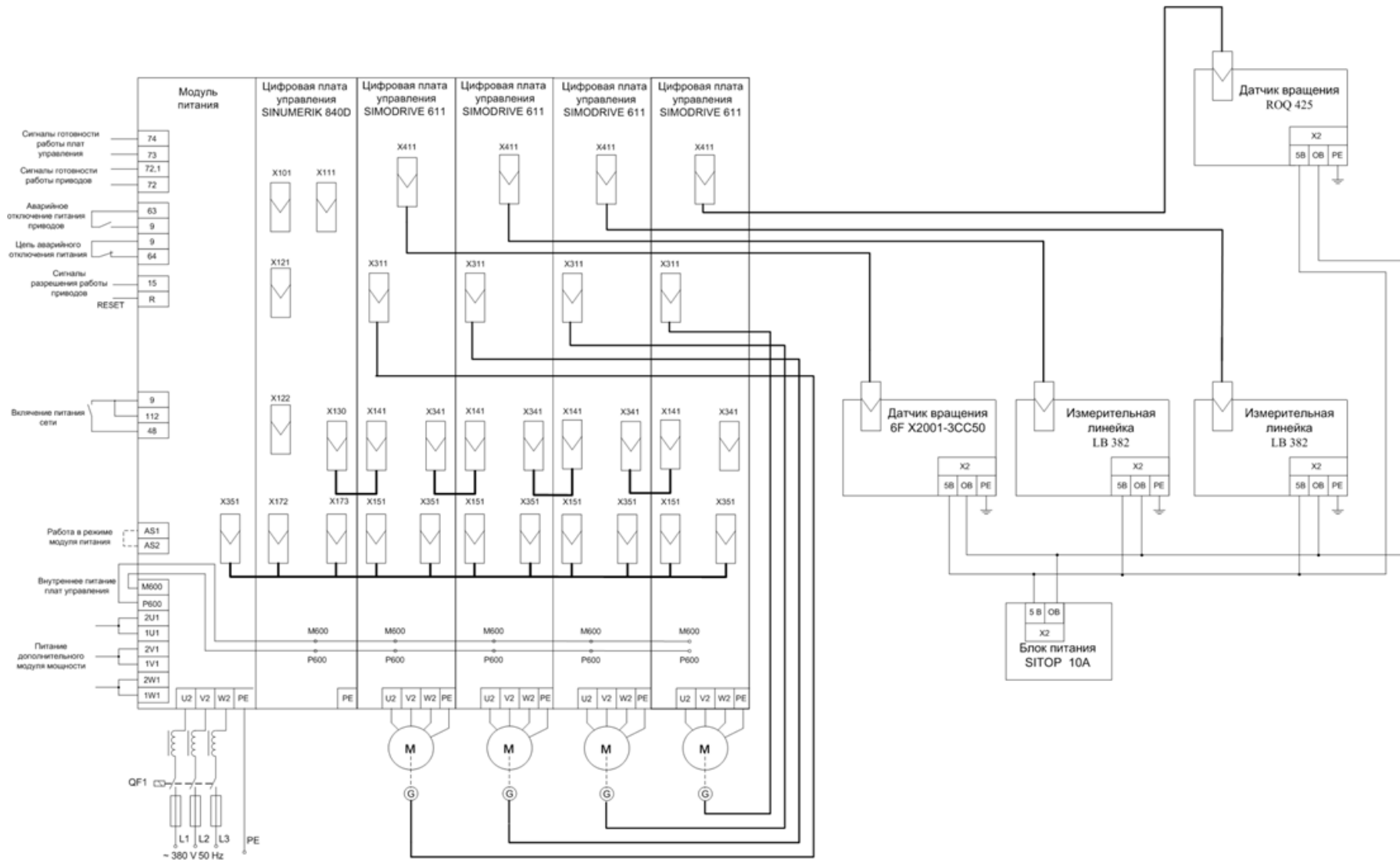


Рисунок 6.14 – Приводна система SINUMERIK 840D для фрезерного верстата із ЧПУ

7 ОРГАНІЗАЦІЯ ПРИВІДНОЇ СИСТЕМИ SINAMICS

7.1 Особливості приводів SINAMICS

У верстатобудуванні й виробництві комплектного промислового устаткування часто використовуються погоджені приводи, що вирішують спільне завдання щодо руху. Для широкого спектра промислових додатків концерн SIEMENS пропонує нову приводну систему SINAMICS. Її функціональна схема наведена на рисунку 7.1.

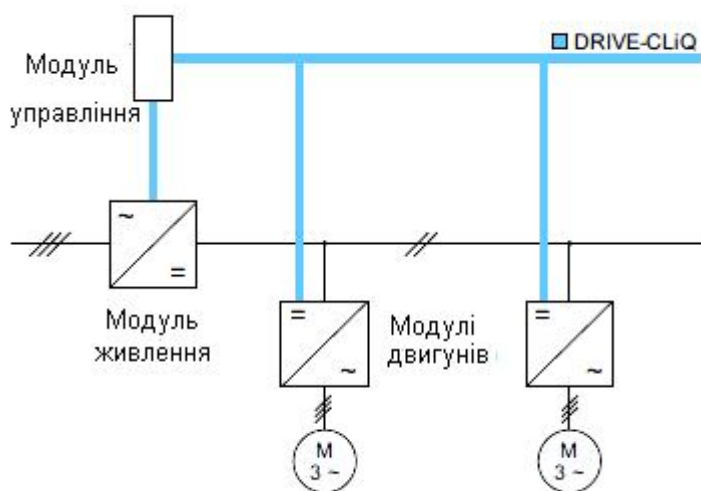


Рисунок 7.1 – Функціональна схема приводної системи SINAMICS

Силові частини вибираються згідно з вимогами до споживаної енергії на рухи робочої машини. Керуючий модуль вибирається залежно від числа регульованих приводів і необхідних робочих характеристик.

Силова частина складається з модуля живлення й модулів двигунів. Модуль живлення виробляє з напруги мережі постійну напругу для живлення модулів двигунів. Модулі двигунів з'єднані загальною шиною постійного струму, що утворює проміжний контур. Завдяки загальному проміжному контуру постійної напруги можливий баланс енергії між модулями двигунів, тобто якщо один модуль двигуна генерує енергію (генераторний режим), то вона може використовуватися іншим модулем двигуна в моторному режимі. Модулі двигунів дозволяють створити багатоосеву систему приводів з керуванням від NCU (Numerical Control Unit) SINUMERIK або керуючого модуля SINAMICS, наприклад, CU320. Для передачі сигналів керування й сигналів зворотного зв'язку використовується спеціально створений інтерфейс DRIVE-CLIQ.

SINAMICS є складовою частиною Комплексної автоматизації (Totally Integrated Automation) від Siemens.

Сумісність SINAMICS у проектуванні, керуванні даними й комунікації з рівнем автоматизації забезпечується системами керування SIMATIC,

SIMOTION і SINUMERIK. У якості комунікаційного з'єднання – залежно від типу перетворювача – пропонуються різні шини:

- PROFINET;
- PROFIBUS;
- AS-interface;
- Canopen;
- Modbus RTU.

Залежно від мети використання, можна вибрати оптимальний перетворювач частоти й інтегрувати його в концепцію автоматизації. Для цього перетворювачі підрозділяються по меті використання:

- Для SINAMICS G в основі стоїть ефективність. Перетворювачі розроблені для розв'язку стандартних завдань із асинхронними двигунами. Варіанти керування починаються з U/f і закінчуються векторним керуванням.
- Для SINAMICS S пріоритетом є гнучкість. Перетворювачі вирішують спеціальні завдання керування синхронними й асинхронними двигунами й відповідають найвищим вимогам по динаміці й точності, інтеграції численних технологічних функцій у регулятори приводів.
- SINAMICS DCM це привод постійного струму. Завдяки своїй універсальній здатності до розширення він відповідає як базовим, так і підвищеним вимогам в області приводної техніки.

У перетворювачах SINAMICS можуть бути реалізовані різні типи застосувань:

- SINAMICS G110 – модульний одиночний привод потужністю від 0,12 кВт до 3,0 кВт і SINAMICS G110D потужністю від 0,75 кВт до 7,5 кВт, невисока точність по швидкості й моменту (насоси, вентилятори, компресори, транспортери та ін.);

- SINAMICS G120, SINAMICS G120C, SINAMICS G120P, SINAMICS G120D – модульний одиночний привод потужністю від 0,37 кВт до 250 кВт, середня точність по швидкості, моменту й положенню (ескалатори, млини, екструдери, суднові приводи та ін.);

- SINAMICS G130 – одиночний привод великої потужності до 560 кВт, виконання «шасі»;

- SINAMICS G150 – одиночний привод великої потужності до 1500 кВт, шафового виконання,

- SINAMICS S110 – модульний одиночний привод потужністю від 0,37 кВт до 90 кВт для виконання переривчастих рухів з невеликими вимогами до точності підтримки моменту, швидкості й позиціонування (складські підйомники, крокові транспортери, пакувальні й складальні агрегати);

- SINAMICS S120 – багатоосевий привод для застосувань з високими вимогами до точності підтримки моменту, швидкості й позиціонування. Цей

тип найбільш прийнятний для побудови приводних систем верстатів із ЧПУ в комбінації з керуючими пристроями SINUMERIK 802D sl, SINUMERIK 828D sl і SINUMERIK 840D sl.

7.2 Приводи SINAMICS для верстатів із ЧПУ

Найпростішою модифікацією SINAMICS S120, призначеної для **одноосьового привода**, є блоковий формат. Конструктивно блоковий формат складається із керуючого модуля CU310-2 (або адаптера керуючого модуля CUA31/CUA32) і силового модуля PM340. Силовий модуль містить у собі мережний випрямляч та інвертор для живлення двигуна. На рисунку 7.2 показані різні типорозміри силових модулів PM340 для двигунів потужністю від 0,12 кВт до 90 кВт.



Рисунок 7.2 – Силові модулі PM340 різної потужності для одноосьових приводів блокового формату

Для **багатоосевих приводів** існують наступні формати систем.

1. SINAMICS S120 Combi – комбінований силовий модуль на 3-4 двигуна змінного струму. Його зовнішній вигляд показано на рисунку 7.3.

SINAMICS S120 Combi



Рисунок 7.3 – Зовнішній вигляд SINAMICS S120 Combi

SINAMICS S120 Combi – це компактна приводна система для невеликих токарських і фрезерних верстатів (шпиндель до 16 кВт і 2-3 привода подачі по 4-6 кВт). Для роботи SINAMICS S120 Combi необхідний пристрій числового керування NCU 710.3 PN, що входить у систему ЧПУ, наприклад, SINUMERIK 828D sl. Приводна система SINAMICS S120 Combi поєднує живильну магістраль із підтримкою рекуперації, а також силові частини для двигунів.

2. «Книжковий» формат. Пристрої «книжкового» формату оптимізовані для багатоосевих додатків і монтуються безпосередньо друг поруч із другом. Тут виділяють конструкцію з *компактним* розв'язком (потужність від 0,9 кВт до 9,7 кВт) і *стандартну* конструкцію (потужність від 1,6 кВт до 107 кВт). Варіанти обох конструкцій показано на рисунку 7.4.



Рисунок 7.4 – Книжковий формат SINAMICS S120 (ліворуч Compact)

3. Формат «шасі». Ці пристрої поставляються як модулі живлення й модулі двигунів і призначені для двигунів великої потужності – від 110 кВт до 1200 кВт.

Загальний вид приводної системи SINAMICS S120 формату «шасі» показано на рисунку 7.5

7.3 Інтерфейс DRIVE-CLIQ

Для всіх варіантів багатоосевих приводів SINAMICS S120 керування приводами верстата здійснюється від пристрою числового програмного керування SINUMERIK, що містить ЧПУ-комп'ютер, наприклад, NCU 710.3 PN.

Цей комп'ютер підключаються до приводної групи через цифровий інтерфейс DRIVE-CLIQ. При цьому максимальне видалення NCU від приводної групи може досягати 100 м.



Рисунок 8.5 – Система SINAMICS S120 формату «шасі»

DRIVE-CLIQ – новий протокол обміну Siemens для систем керування рухом SINAMICS. Це базований на Ethernet інтерфейс для зв'язку модулів керування із силовими модулями двигунів і датчиками положення. На рисунку 7.6 наведені позначення контактів рознімання кабелю DRIVE-CLIQ.

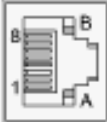
	Контакт	Имя сигнала	Технические данные
	1	TXP	Передаваемые данные +
	2	TXN	Передаваемые данные -
	3	RXP	Принимаемые данные +
	4	зарезервировано, не использовать	
	5	зарезервировано, не использовать	
	6	RXN	Принимаемые данные -
	7	зарезервировано, не использовать	
	8	зарезервировано, не использовать	
	A	+ (24 В)	Электропитание
	B	M (0 В)	Масса электроники

Рисунок 7.6 – Распиновка рознімання кабелю DRIVE-CLIQ

Незважаючи на те, що в цей час є широкі можливості для вибору датчиків з інтерфейсом DRIVE-CLIQ, існує можливість використання датчиків, що не мають інтерфейсу DRIVE-CLIQ.

Для таких випадків пропонуються **модулі датчиків** зі ступенем захисту IP20 для монтажу в електрошафи (зовнішній вигляд показано на рисунку 7.7) і модулі датчиків зі ступенем захисту IP67, які можуть бути розташовувані на встаткуванні разом з датчиком (зовнішній вигляд показано на рисунку 7.8).



Рисунок 7.7 – Загальний вид модуля датчика для монтажу в шафі



Рисунок 7.8 – Загальний вид модуля датчика для монтажу поза шафою

Модулі датчиків, які монтуються в шафі, можуть обробляти сигнали практично всіх типів датчиків.

Модуль SMC10 обробляє сигнали резольвера.

Модуль SMC20 обробляє сигнали інкрементального датчика \sin/\cos 1 Vpp і сигнали абсолютного датчика з інтерфейсом Endat.

Модуль SMC30 обробляє сигнали інкрементальних датчиків TTL/HTL і сигнали TTL/HTL абсолютних датчиків послідовного синхронного інтерфейсу SSI. Модуль має наступні стандартні з'єднання:

- з'єднання інтерфейсу DRIVE-CLiQ;
- з'єднання з датчиком, сигнали якого перетворюються;
- з'єднання з датчиком температури двигуна (КТУ84-130 або РТС) через штекер Sub-D або клеми;
- з'єднання із джерелом живлення електроніки через штекер DC 24 В;
- підключення РЕ/захисного проведено.

На рисунку 7.9 наведений приклад використання модулів SMC.

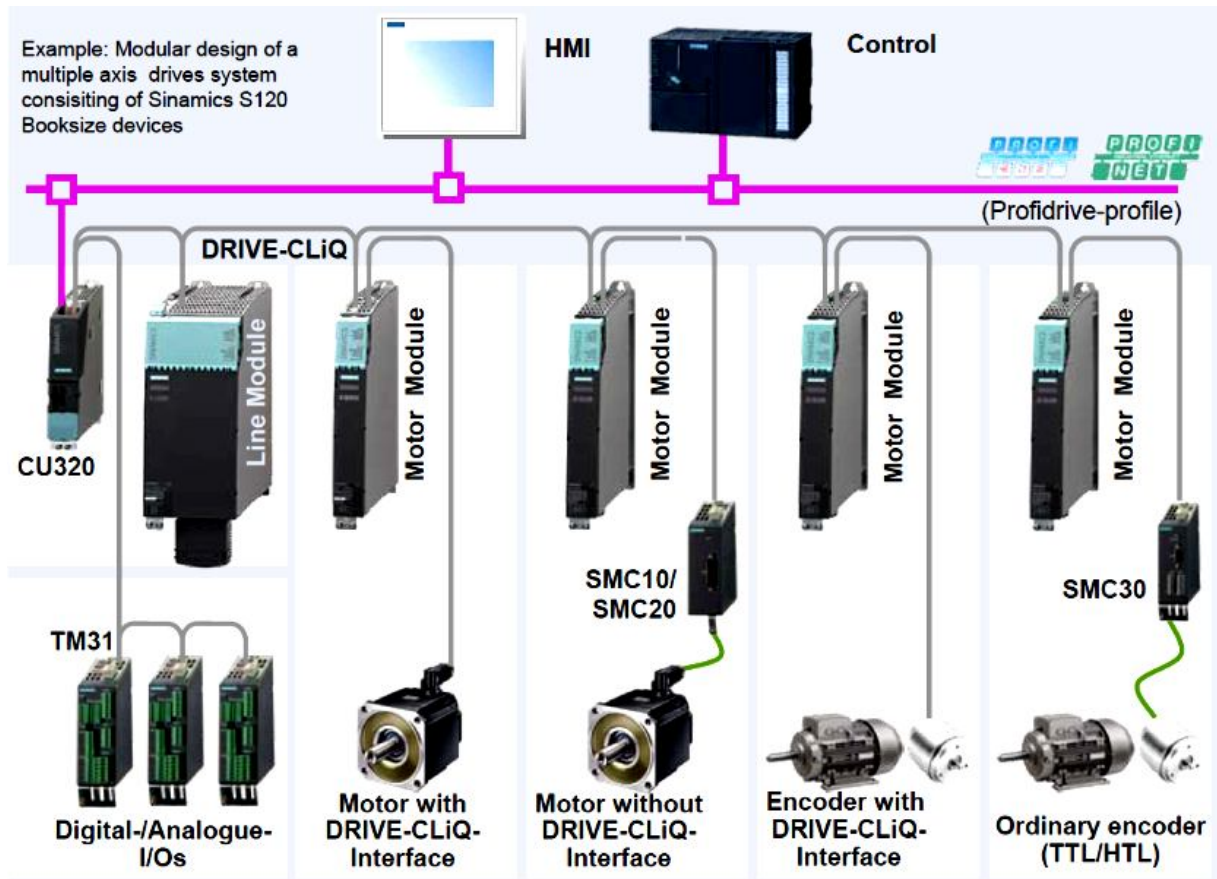


Рисунок 7.9 – Приклад системи SINAMICS S120, у якій інтерфейс DRIVE-CLiQ створений модулем управління CU320

Зовнішні модулі датчиків SME20/SME25 – це блоки обробки сигналів прямих вимірювальних систем верстатів. Вони мають корпус зі ступенем захисту IP67, завдяки чому блоки можуть монтуватися *поза електрошафою поблизу від датчика*.

Модуль SME20 обробляє сигнали інкрементального датчика sin/cos 1 Vpp без тракту C/D відстеження положення ротора. Модуль SME25 обробляє сигнали абсолютного датчика Endat або абсолютного датчика типу SSI з інкрементальними сигналами sin/cos 1 Vpp)

Максимальна довжина кабелю між датчиком і модулем SME20/SME25 становить 3 м. Максимальна довжина кабелю між модулем SME20/SME25 і силовим модулем привода становить 100 м.

Модулі SME20/SME25 мають наступні інтерфейси:

- інтерфейс DRIVE-CLIQ;
- інтерфейс для датчика;
- 24 В DC (живлення датчика від модуля двигуна);
- PE рознімання для заземлення.

7.4 Вимірювальні системи з інтерфейсом DRIVE-CLIQ

Розрізняють інкрементальний і абсолютний методи виміру. Датчики, використовувані для цих вимірів, називаються енкодерами, тобто пристроями, що дозволяють одержати *n-розрядний* код вимірюваної величини.

Енкодер з інтерфейсом DRIVE-CLIQ передає дані про положення й швидкості по одному протоколу.

Одна із ключових переваг енкодера з інтерфейсом DRIVE-CLIQ – це значне спрощення *настроювання й конфігурації* при установці енкодера в системі. Кожна одиниця має електронну мітку з базовими даними, такими як серійний номер і технічна специфікація. Завдяки цим даним енкодер визначається системою при підключенні й автоматично настраюється згідно з його конфігурацією.

Інша перевага енкодерів з інтерфейсом DRIVE-CLIQ – це скорочення асортименту *кабельної продукції*. Інтерфейс дозволяє використовувати загальний кабель, до якого з використанням хаба можуть бути підключені до п'яти енкодерів. Це значно скорочує витрати й час на монтаж системи.

Як відмічалось вище, у системах автоматизації верстатів існують інкрементальні й абсолютні енкодери. Характеристики деяких з них наведено на рисунку 7.10.

Інкрементальні енкодери простіше по конструкції в порівнянні з абсолютними енкодерами. На оптичному диску такого енкодера є зазвичай три доріжки. Енкодери працюють за принципом оптоелектронного розгорнення ділильних дисків у минаючій світлі. Джерелом світла є світлодіод (LED).

Виникаюча при обертанні вала енкодера модуляція світлотіні реєструється фотоелементами. За допомогою погодженого розташування штрихового зразка на з'єднаному з валом ділильному диску й зафіксованої діафрагми, фотоелементи посилають два зміщені по відношенню друг до друга на 90° подорожніх сигналу А і В, а також нульовий сигнал R. Електроніка енкодера підсилює ці сигнали й перетворює їх у різні вихідні інтерфейси.




Тип датчика	Интерфейс	Safety Integrated ¹⁾	Точность в угловых секундах	Разрешение	Степень защиты без/с входом вала
Инкрементальные энкодеры 	sin/cos 1 V _{pp}	да	± 18 мех. × 3600/ число делений Z	2500 имп/об	IP67/IP64
	RS422 (TTL)	2)	± 18 мех. × 3600/ число делений Z	5000 имп/об	IP67/IP64
	HTL	2)	± 18 мех. × 3600/ число делений Z	2500 имп/об	IP67/IP64
	RS422 (TTL) двойн. дорожка	2)	дорожка 1: ± 63 дорожка 2: ± 12	дорожка 1: 1024 имп/об дорожка 2: 9000 имп/об	IP67/IP64
Абсолютные энкодеры  	DRIVE-CLiQ	2)	± 36	<u>однооборотный</u> 22 бит <u>многооборотный</u> 34 бит (22 бит однооборотный + 12 бит многооборотный)	IP67/IP64
	SSI	2)	± 79 (при 8192 шагах)	<u>однооборотный</u> 13 бит (8192 шагов) <u>многооборотный</u> 25 бит (8192 шагов × 4096 оборотов)	IP67/IP64
	EnDat	да	± 60 (инкр. дорожка)	<u>однооборотный</u> 13 бит (8192 шагов) <u>многооборотный</u> 25 бит (8192 шагов × 4096 оборотов)	IP67/IP64
	PROFIBUS DP	2)	± 79 (при 8192 шагах)	<u>однооборотный</u> 13 бит (8192 шагов) <u>многооборотный</u> 27 бит (8192 шагов × 16384 оборота)	IP67/IP64
	PROFINET IO	2)	± 79 (при 8192 шагах)	<u>однооборотный</u> 13 бит (8192 шагов) <u>многооборотный</u> 27 бит (8192 шагов × 16384 оборота)	IP67/IP64

Рисунок 7.10 – Типи датчиків і їх характеристики

Існують наступні варіанти вихідних сигналів інкрементальних датчиків:

- Аналогові сигнали sin/cos з рівнем 1 V_{pp} і дозволом 2500 періодів/оберт. Для одержання ще більш високого дозволу сигнали sin/cos цих енкодерів інтерполюються (множаться) при перетворенні у цифровий код.
- Диференціальні імпульсні сигнали відповідно до інтерфейсу RS422 (рівень TTL, напруга живлення 5 В) з дозволом 5000 імп/оберт.
- Диференціальні імпульсні сигнали з датчиків HTL (High Voltage Transistor Logic). Датчики з інтерфейсом HTL призначені для додатків із цифровими входами, які мають рівень 24 В.

Слід прийняти до уваги, що є інкрементальні енкодери з п'ятьма доріжками – А, В, С, D і R, у яких з доріжок С, D знімаються аналогові синусно-косинусні сигнали дозволом один період/оберт для контролю кута повороту ротора двигуна.

Абсолютні енкодери мають ту ж систему зчитування, що й інкрементальні енкодери, але відрізняються більшою кількістю доріжок. Зчитувана з доріжок інформація кодується однокроковим кодом Грея.

Цей код дозволяє уникнути помилок зчитування. В абсолютних енкодерах після включення верстата на кожній доріжці формується позиційне значення, яке передається в систему керування. Тому реферування вимірювальної системи верстата із абсолютними енкодерами не потрібно.

Існують однооборотні й багатооборотні абсолютні енкодери. Однооборотні енкодери мають один оптичний диск із n доріжками й зчитуваний із цього диска код має максимальне значення, рівне $N = 2^n$.

Багатооборотні енкодери на додаток до абсолютного положення в межах одного оберту реєструють і число обертів. Для цього зчитуються інші кодові диски, з'єднані через редуктор з валом енкодера. При обробці, наприклад, 13 доріжок диска для визначення положення в межах одного оберту й 12-ти інших доріжок для лічильника обертів максимальне значення коду досягає $2^{25} = (8192 \text{ кроків} \times 4192 \text{ обертів})$.

7.5 Двигуни для підключення до SINAMICS

У теперішній час Siemens створена лінійка двигунів SIMOTICS для приводних систем SINAMICS.

Сервомотори 1PH8 (асинхронні)

Сервомотори 1PH8 – це нове покоління двигунів для приводів головного руху. Асинхронні сервомотори 1PH8 виконані з **короткозамкненим ротором** і ступенем захисту IP55/IP65. Вони заміняють двигуни 1PH7, які добре зарекомендували себе. Двигуни 1PH8 поставляються із двома різними типами охолодження: примусова повітряна вентиляція й водяне охолодження.

Залежно від завдань регулювання із пропонованих варіантів можна вибрати підходящі датчики для реєстрації швидкості двигуна й непрямого положення робочого вузла верстата.

Сервомотори 1PH8 мають наступні переваги:

- у двигуни вбудовані різні типи датчиків високої точності;
- максимальне число обертів доведене до 20 000 об/хв;
- величина биття не перевищує 10 мкм;
- малий рівень вібрацій;
- висока динаміка (короткий час розгону);
- проста й гнучка техніка з'єднання.

Область застосування:

- складні обробні центри й шпинделі токарських верстатів;
- фрезерні шпинделі для високого навантаження;
- зустрічні шпинделі або обертові інструменти на токарських верстатах, спеціальні верстати;
- інструменти із прямим приводом і внутрішнім охолодженням.

Серводвигуни 1PH8 (синхронні)

Синхронні двигуни 1PH8 із збудженням від постійних магнітів, що мають ступінь захисту IP55/IP65 і розширюють можливості використовуваної сьогодні, надійної серії двигунів типу 1FT.

Двигуни були розроблені спеціально для використання з лінійкою приводів SINAMICS S120. Залежно від завдань регулювання, для даних двигунів є підходящі датчики для реєстрації швидкості й непрямого положення.

Переваги:

- широкий діапазон потужностей;
- різні типи підшипників;
- різні типи датчиків високої точності для регулювання швидкості й позиціонування;
- висока точність обертання, низький рівень вібрації, низький рівень шуму;
- висока динаміка (короткий час розгону).

Двигуни для приводів подач

Двигуни 1FT6 це синхронні електродвигуни із збудженням від постійних магнітів з номінальним моментом від 1,4 Нм до 74 Нм.

Двигуни 1FT7 мають більш компактне виконання при тих же характеристиках.

Двигуни 1FK7 це компактні синхронні електродвигуни, з дуже малим моментом інерції.

На рисунку 7.11 наведені деякі характеристики двигунів подач.



Тип двигателя	Описание	Степень защиты	Тип охлаждения
Серводвигатели SIMOTICS S – синхронные двигатели с возбуждением от постоянных магнитов			
	SIMOTICS S-1FT7 Compact Двигатель подачи – Compact Очень высокая удельная мощность	IP64 ¹⁾ (как опция IP65, IP67)	Самоохлаждение Принудительная вентиляция Водяное охлаждение
	SIMOTICS S-1FT7 High Dynamic Двигатель подачи – High Dynamic Очень низкий момент инерции ротора	IP64 (как опция IP65, IP67)	Принудительная вентиляция Водяное охлаждение
	SIMOTICS S-1FK7 Compact Двигатель подачи – Compact Высокая удельная мощность Compact для силовых модулей 1 AC 230 В	IP64 (как опция IP65)	Самоохлаждение
	SIMOTICS S-1FK7 High Dynamic Двигатель подачи – High Dynamic Очень низкий момент инерции ротора High Dynamic для силовых модулей 1 AC 230 В	IP64 (как опция IP65)	Самоохлаждение
	SIMOTICS S-1FK7 High Inertia Двигатель подачи – High Inertia Высокая или переменная инерция нагрузки	IP64 (как опция IP65)	Самоохлаждение

Рисунок 7.11 – Загальний вид і деякі характеристики двигунів подач

7.6 Модулі живлення для приводів

Модулі живлення виробляють із трифазної мережі напругу постійного струму й живлять модулі двигунів через проміжний контур. У сімействі SINAMICS S120 є наступні типи модулів живлення: модуль живлення Basic для формату «шасі», нерегульований модуль Smart і регульований активний (Active) модуль живлення для книжкового формату.

Модулі живлення Smart

Модулі живлення Smart через нерегульований перетворювач струму подають енергію в проміжний контур постійного струму приводної групи, а також можуть рекуперувати зайву генераторну енергію приводної групи в мережу. Схема живлення наведена на рисунку 7.12.

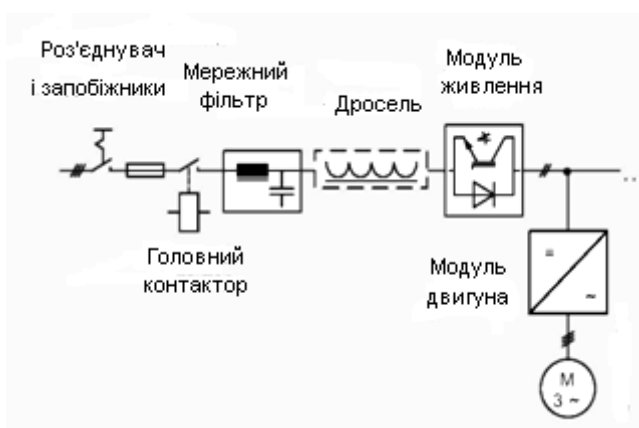


Рисунок 7.12 – Схема живлення двигуна із модулем Smart

Недоліком цієї схеми є те, що коливання напруги мережі пропорційно переходять на напругу проміжного контуру і далі на фазні обмотки двигунів.

При використанні модуля живлення Smart необхідний відповідний мережний дросель. Як опція для демпфірування напруги перешкод може бути встановлений мережний фільтр для дотримання граничних значень згідно зі стандартом EN 61800-3.

Модуль гальмування й гальмовий резистор необхідні, якщо потрібне цілеспрямоване гальмування приводів.

Активні модулі живлення

Активні модулі живлення можуть живити шину проміжного контуру енергією й рекуперувати надлишкову генераторну енергію в мережу. На відміну від модулів живлення Smart, активні модулі живлення виробляють регульовану напругу постійного струму, яка підтримується постійною незалежно від коливань напруги в мережі. При цьому напруга в мережі повинна перебувати в межах дозволених допусків.

Модулі гальмування й гальмовий резистор необхідні тільки в тому випадку, якщо при відключенні живильної мережі без можливості рекуперації

потрібне цілеспрямоване гальмування приводів.

Активні модулі живлення одержують із мережі практично синусоїдальний струм і тому не викликають гармонік високої частоти. Обов'язковою умовою є наявність *активного інтерфейсного модуля* відповідної потужності. Схема з'єднання активного інтерфейсного модуля й модуля живлення показана на рисунку 7.13.

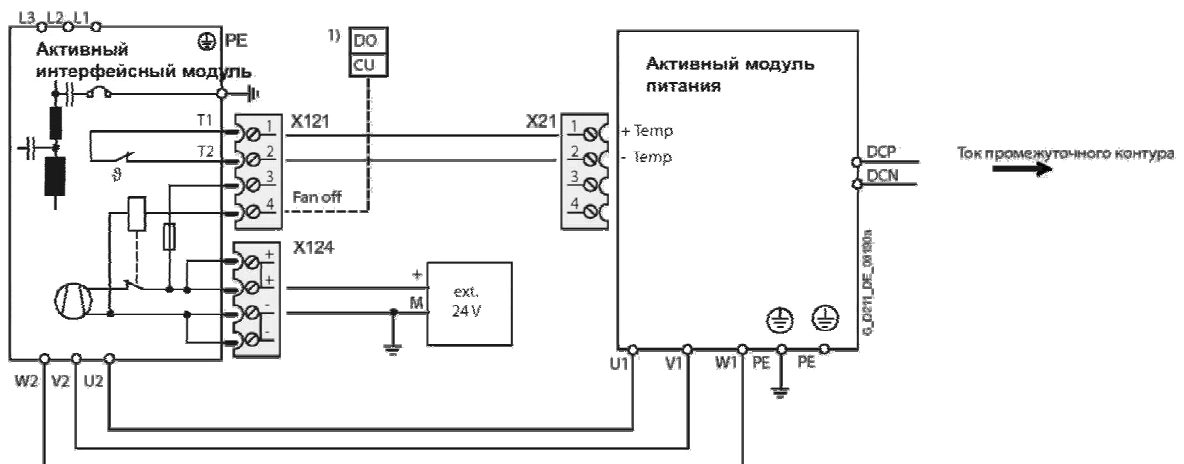


Рисунок 7.13 – Схема активного модуля живлення

Активні інтерфейсні модулі утворюють загальний функціональний вузол з активними модулями живлення і є обов'язковою умовою для їхньої роботи. Активні інтерфейсні модулі обладнані фільтром Clean Power і придушують перешкоди, відповідаючи тим самим категорії С3 згідно EN 61800-3 по емісії перешкод.

Фільтр Clean Power видаляє гармоніки, викликані частотою перемикачання, з живильної мережі. Завдяки цьому в приводну систему з живильної мережі надходить синусоїдальний струм і практично не створюються вищі гармоніки.

7.7 Уведення й вивід сигналів автоматики

Для введення-виводу сигналів автоматики можна використовувати термінальні модулі ТМ15 і ТМ31. За допомогою термінального модуля ТМ15 можна збільшити число наявних цифрових входів/виходів у приводній системі.

Зовнішній вигляд модулів ТМ15 і ТМ31 показано на рисунку 7.14.



Рисунок 7.14 – Зовнішній вигляд модуля TM15 (ліворуч) і модуля TM31 (праворуч)

На термінальному модулі TM15 перебувають:

- 24 двунправлених цифрових входу/виходу (розв'язка по напрузі на 3 групи по 8 каналів у кожній);
- 24 зелених LED для індикації логічного стану сигналу відповідної клеми;
- 2 рознімання DRIVE-CLIQ;
- 1 рознімання для живлення електроніки через штекер живлення DC 24 В;
- 1 підключення РЕ/захисного кабелю.

Термінальний модуль TM15 може бути змонтований на DIN-рейку TH 35 по EN 60715.

Стан термінального модуля TM15 відображується через багатобарвний LED.

Термінальний модуль TM15 зв'язується через DRIVE-CLIQ із керуючим модулем CU320 або з керуючим модулем SINUMERIK sl.

За допомогою термінального модуля TM31 можна збільшити число наявних цифрових входів/виходів, а також аналогових входів/виходів у приводній системі.

Крім цього, термінальний модуль TM31 має релейні виходи з перемикаючим контактом і вхід датчика температури.

На термінальному модулі TM31 перебувають:

- 8 цифрових входів;
- 4 двунправлених цифрових входу/виходи;
- 2 релейних виходи з перемикаючим контактом;
- 2 аналогових входи;
- 2 аналогових виходи;
- 1 вхід датчика температури (КТУ84-130 або РТС);

- 2 рознімання DRIVE-CLIQ;
- 1 рознімання для живлення електроніки через штекер DC 24 В;
- 1 підключення РЕ/захисного кабелю.

Термінальний модуль TM31 може бути змонтований на DIN-рейку TH 35 по EN 60715.

7.8 Порядок проектування приводної системи SINAMICS S120

Проектування починається з вибору механічного зв'язку з робочою машиною. Далі визначаються необхідні крутні моменти й швидкості, вибираються відповідні силові компоненти. Згідно із вимогами технології двигун може житись як індивідуальний привод від силового модуля або як частина багатоосевого привода від модуля двигуна.

Після вибору базових компонентів відбувається вибір інших системних компонентів для адаптації привода до електричних і механічних умов.

Процес проектування здійснюється в програмному середовищі SIZER у такій послідовності.

1. Вибір виду привода.
2. Визначення розрахункових навантажень, розрахунки максимального моменту навантаження, визначення необхідної потужності двигуна.
3. Вибір модулів двигунів.
4. Вибір датчиків.
5. Повторення кроків 3 і 4 для всіх двигунів.
6. Розрахунки необхідної потужності проміжного контуру й вибір модуля живлення.
7. Визначення необхідної продуктивності системи регулювання й вибір модуля керування, проектування схеми з'єднань компонентів.
8. Визначення силових компонентів з боку мережі (вимикач, запобіжники, мережні фільтри і т.д.).
9. Обчислення споживання струму DC-24 В всіх компонентів і вибір блоку живлення (SITOP, модуль живлення електроніки).
10. Вибір сполучних кабелів.
11. Розрахунки необхідних поперечних перерізів кабелів для підключення мережі й двигуна.

ЛИТЕРАТУРА

1. Сосонкин, В. Л. Системы числового программного управления: Учеб. пособие / В. Л. Сосонкин, Г. М. Мартинов – М.: Логос, 2005. – 296 с.
2. Ловыгин А. А. Современный станок с ЧПУ и САД/САМ-системы / А.А. Ловыгин, А.В. Васильев, С.Ю. Кривцов. – М. : «Эльф ИПР», 2006. – 286 с.
3. SIMORIVE. Платы управления: Документация Siemens. – Выпуск 06/2004.
4. Сосонкин, В. Л. Методика программирования станков с ЧПУ / В. Л. Сосонкин, Г. М. Мартинов : [Электронный ресурс] Режим доступа: <http://www.ncsystems.ru/ru/education/lectures/posu>
5. Siemens NC 61 2010. Приводная система **SINAMICS S120** [Электронный ресурс] Режим доступа: http://www.power-prom.ru/docs/06_NC61_2010.pdf

