

# PROGRAMAREA ECHIPAMENTELOR CNC

---

Liviu Morar

Emilia Câmpean



Editura U.T.PRESS  
Str.Observatorului nr. 34  
C.P.42, O.P. 2, 400775 Cluj-Napoca  
Tel.:0264-401.999 / Fax: 0264 - 430.408  
e-mail: [utpress@biblio.utcluj.ro](mailto:utpress@biblio.utcluj.ro)  
[www.utcluj.ro/editura](http://www.utcluj.ro/editura)

Director: Ing. Călin D. Câmpean

Copyright © 2015 Editura U.T.PRESS

Reproducerea integrală sau parțială a textului sau ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al editurii U.T.PRESS.

**ISBN 978-606-737-081-2**

Bun de tipar: 17.09.2015

# Obiective generale

- Suportul de curs: “**Programarea echipamentelor CNC**” se adresează studenților și specialiștilor din domeniul utilizării mașinilor cu comandă numerică
- Ob.1. Să familiarizeze pe cei interesați cu modalitățile de programare: manuală (limbaj ISO)  
asistată de calculator (CAM)
- Ob. 2. Să poată redacta programe de prelucrare în 2  
 $\frac{1}{2}$  axe
- Ob.3. Să poată configura echipamentul numeric pentru implementarea programului scris (alegerea sculelor, parametrilor de așchiere, declararea originii piesei, alegerea sistemului de prindere, etc.).



**Necesită:**

cunoștințe medii în domeniul fabricației (tolerante. m-u, tehnologii).

**Cadru didactic:**

Prof.Dr.Ing. Liviu Morar

Șef Lucrări Dr.Ing. Emilia Câmpean

Birou: Cluj-Napoca, B-dul Muncii, Tel.: 0264 401762

Laborator : E07

**Cărți:**

*Programarea sistemelor numerice de Liviu Morar;*

*Mașini și instalații în sisteme robotizate de C.Pop, L.Morar, M.Galiș;*

*Bazele Programării numerice a mașinilor-unelte, de Liviu Morar;*

*Sisteme integrate de prelucrare, vol.1., de Liviu Morar.*

**Descrierea cursului:**

Studiu referitor la principiile, tehnicile și aplicațiile CNC. Metode de programare (manuală și asistată) a m-u, sisteme de scule.

Prezentarea funcționării echipamentelor numerice este abordată în volumul Echipamente Numerice.



# Bibliografie recomandată:

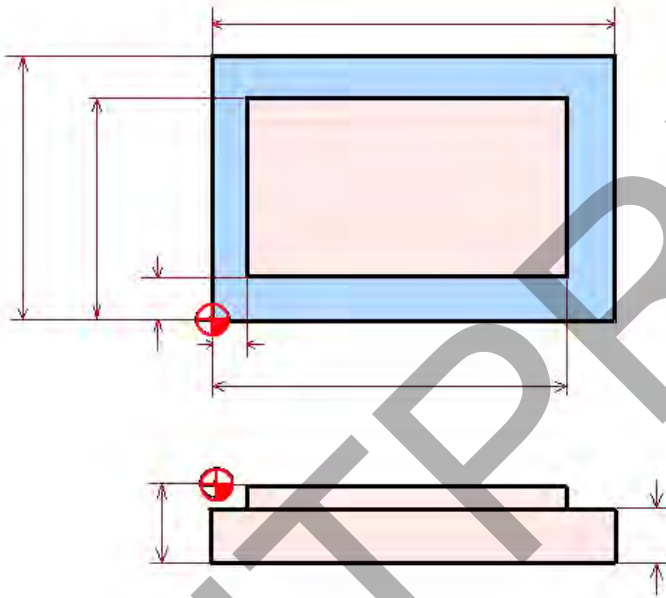
- 1). MORAR, L. *Programarea sistemelor numerice CNC*, Editura UTPRES, 2006
  - Cap.1. Introducere
  - Cap.2. Arhitectura unui program NC
  - Cap.3. Programarea manuala CNC
  - Cap.4. Programarea deplasarilor
  - Cap.5. Informatii tehnologice
  - Cap.6.Cicluri fixe
  - Cap.7.Subprograme
- 2). MORAR, L. *Bazele programării numerice*, Editura UTPRES, 2005
- 3). MORAR, L. ș.a. *Sisteme integrate de prelucrare, Volumul -I- , Bazele sistemelor integrate de prelucrare*, Editura DACIA, 1998
  - Cap.8. Programarea strungurilor
- 4). POP, C. AL., MORAR, L., GALIȘ, M., *Mașini și instalații în sisteme robotizate*  
Editura DACIA, 1999
- 5). MORAR,L., BREAZ, R., CAMPEAN, E., *Programarea manuală și asistată de calculator a echipamentelor numerice*, Casa Cărții de Știință, 2014
  - Cap.9. Computer aided manufacturing



# Tema 1. Program pentru frezare (Nr.1)

Nume \_\_\_\_\_

Scrieți un program, centru de prelucrare Microcut, pentru piesa următoare:



Piesa este realizată dintr-un aliaj de aluminiu.

Prelucrarea se va face în 2 faze:

- degroșare, adaos pentru finisare pe contur 2 mm;
- finisare;

Prelucrarea se va realiza cu aceeași sculă.

**Se cere:**

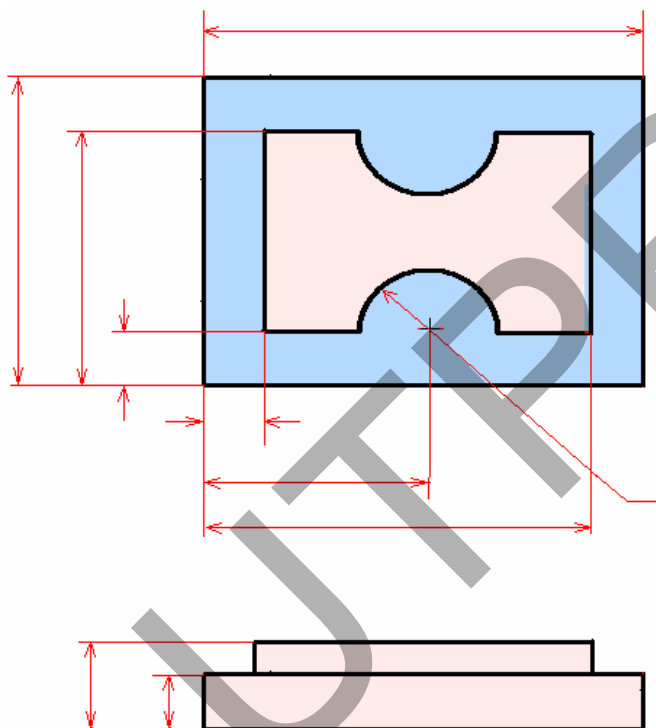
- explicați fiecare linie din program;
- calculați regimul de așchiere;
- indicați traiectoria centrului sculei;
- explicați alegerea sculei;



# Tema 2. Program pentru frezare (Nr.2)

Nume \_\_\_\_\_

Scrieți un program, centru de prelucrare Microcut, pentru piesa următoare:



Piesa este realizată dintr-un aliaj pe bază de aluminiu.

Prelucrarea se va face în 2 faze:

- degroșare, adaos pentru finisare pe contur 2 mm;
- finisare;

Prelucrarea se va realiza cu aceeași sculă.

Se cere:

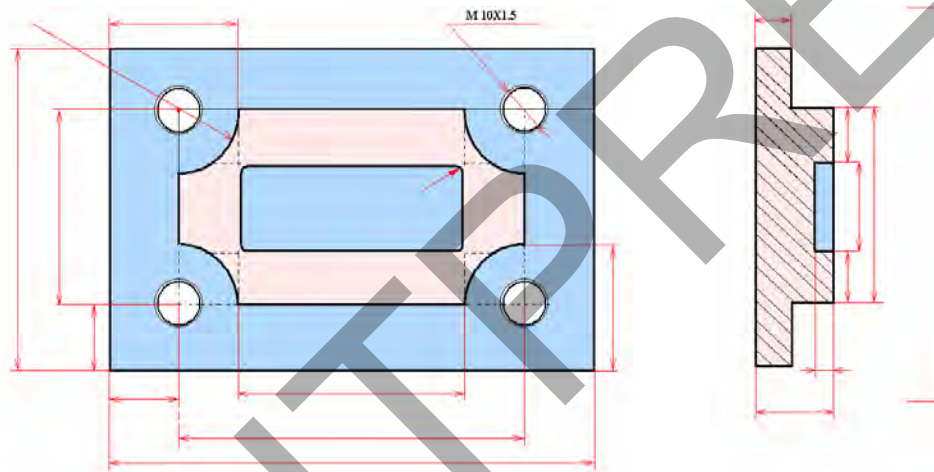
- explicați fiecare linie din program;
- calculați regimul de așchiere;
- indicați traiectoria centrului sculei;
- explicați alegerea sculei;

# Tema 3. Program pentru frezare (Nr.3 și Nr.4)

Nume \_\_\_\_\_

Scrieți 2 programe, pe mașina Microcut, pentru piesa următoare.

- Programul Nr. 3 se întocmește pentru realizarea piesei din figură cu excepția găurilor filetate M10x1,5.
- Programul Nr. 4 este destinat celor 4 găuri filetate.



Se cere:

- explicați fiecare linie din programe;
- calculați parametrii regimului de așchiere, cu indicare clară a documentării, titlu carte, tabel, pagină etc;
- explicați alegerea sculei;

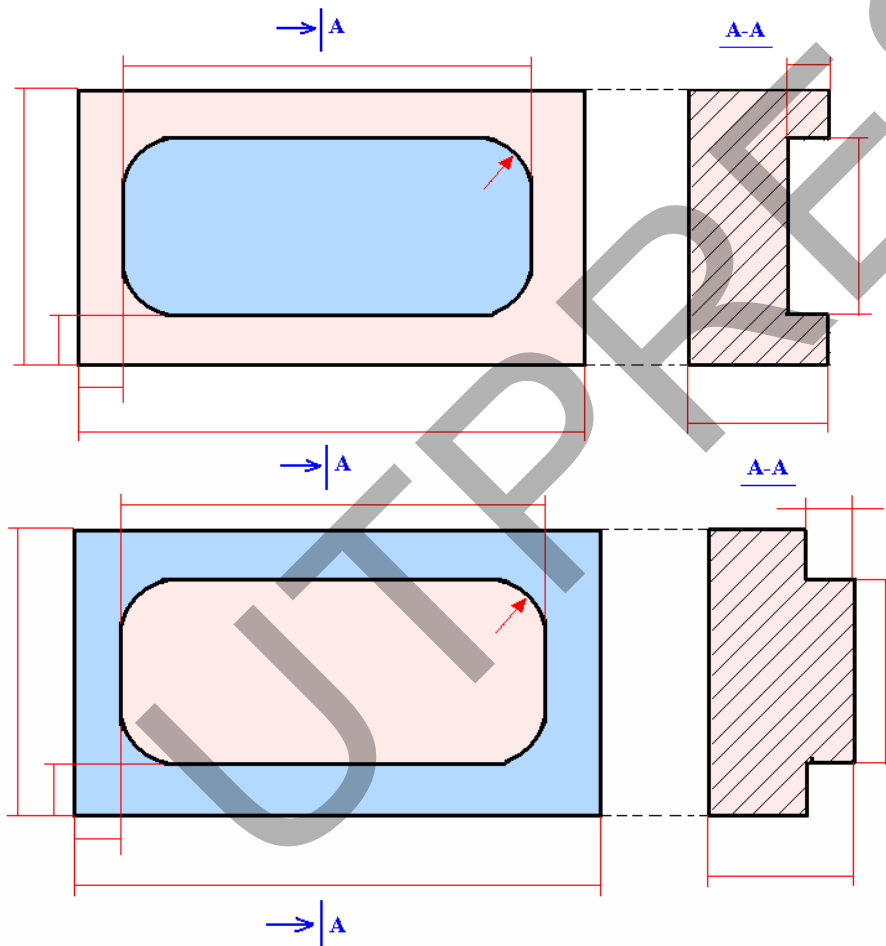




# Tema 4. Program pentru frezare (Nr.5 și Nr.6)

Nume \_\_\_\_\_

Scrieți 2 programe, pe mașina Microcut, pentru piesele următoare.



Se cere:

- explicați fiecare linie din programe;
- calculul parametrilor tehnologici (cu indicarea exactă a sculelor)
- explicați alegerea sculelor.

Eventual sugerați și o altă posibilitate.



# Introducere

## Repere în dezvoltarea comenzii numerice

1955 John Parsons și US Air Force formulează necesitatea dezvoltării unei mașini-unelte capabilă să realizeze piese complexe în condiții de înaltă precizie pentru industria aeronautică cu păstrarea calității în timp (repetabilitate). Subcontractor MIT.

1959 MIT anunță limbajul de programare APT (Automatic Programmed Tools).

1960 DNC (Direct Numerical Control). Elimină utilizarea benzii perforate ca suport de introducere a programului sursă. Fișierele se transmit direct de la calculator la echipamentul numeric.



## Cap.1. Introducere

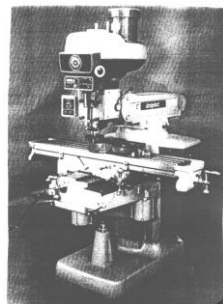
- 1968 Kearney & Trecker produc primul centru de prelucrare.
- Anii '70 Mașini-unelte CNC. Control numeric distribuit.
- Anii '80 Introducerea sistemelor CAM. Sistemele UNIX și bazate PC sunt accesibile.
- Anii '90 Căderea prețului în tehnologia CNC.
- Anii '97 Introducerea sistemului OMAC (Open Modular Architecture Control) pentru înlocuirea sistemelor numerice hardware.



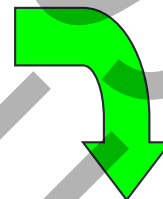
## Control Numeric (Numerical Control)

### Esența CN

Din punct de vedere a m-u



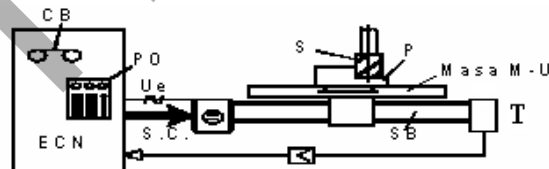
Apariția CNC



### Evoluția structurii M – U

- M U convenționale.

Mișcări controlate prin roți de manevră, mânere, etc.



- M U C N

Mișcări controlate de ECN, mânerle înlocuite prin motor.

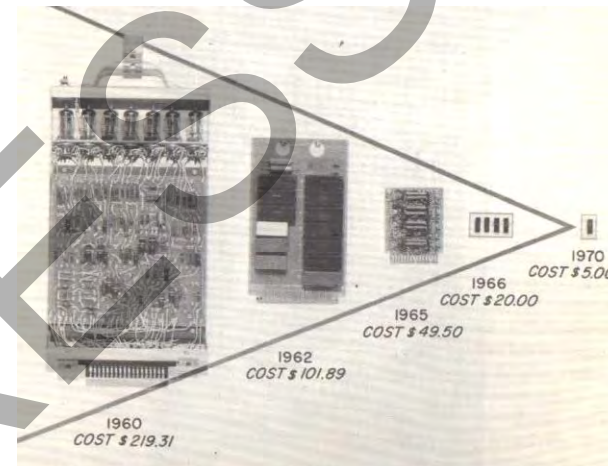
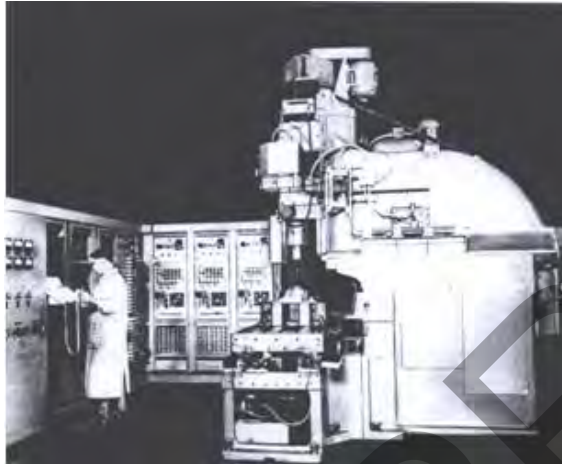
## Din punct de vedere a comenzii



Mașina acționată automat prin comenzi codate pe un mediu digital.



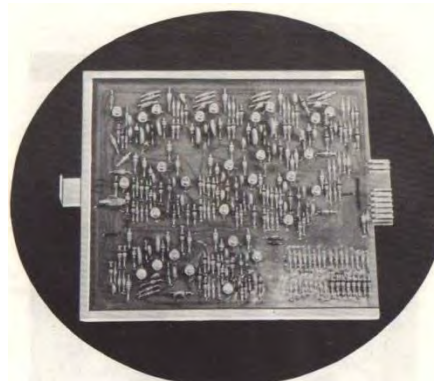
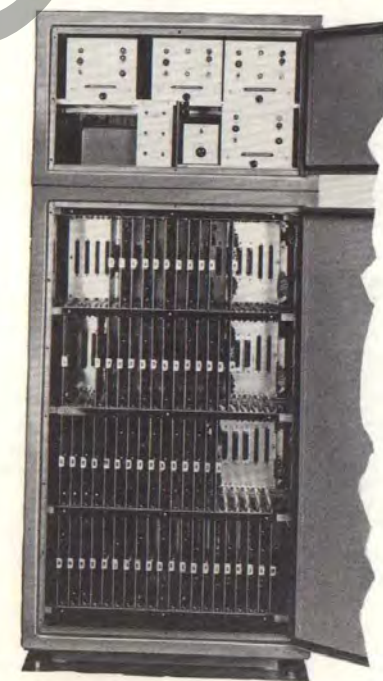
## Prima m-u cu comandă numerică



## Succint istoric

- nevoia de piese precise pentru US Air Force și John Parson – președintele companiei Parsons Works of Traverse City (Michigan) stau la baza primei MUCN.
- proiectul Parsons a fost implementat în laboratorul de Servo Mecanisme din MIT. General Motors dezvoltă simultan traductoare de poziție.
- servo sistem: preia date înregistrate pentru a le produce de una, sau mai multe ori. Tehnica aceasta este denumită record/playback - o reminiscentă a pieselor mecanice.
- (Nuvela Pianistul – de Kurt Vonnegut a fost inspirată de o mașină. Publicist GE).

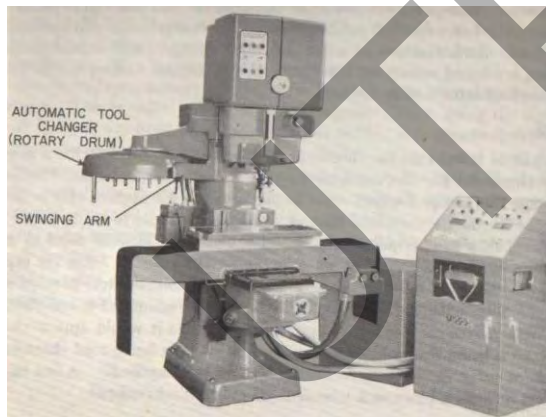
## Primele variante de echipamente NC



# Caracteristici principale ale mașinilor CNC:

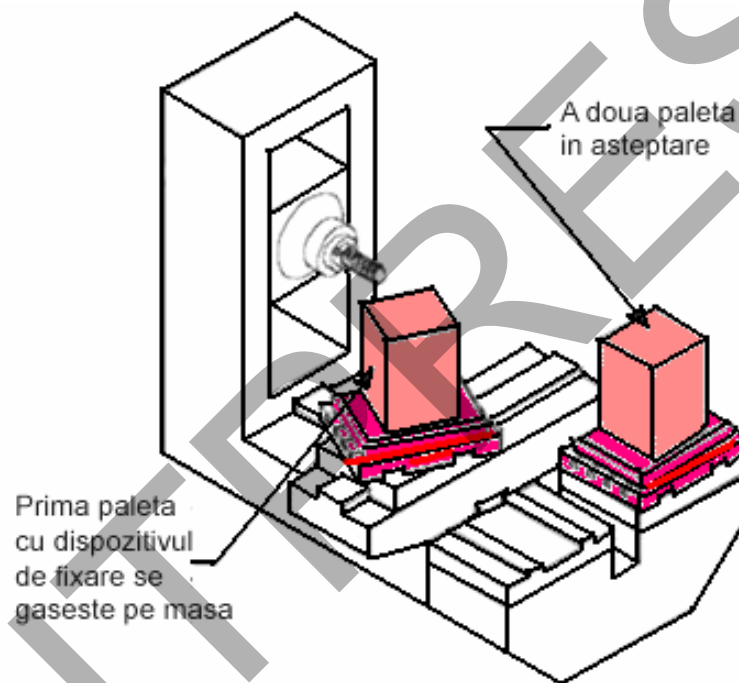
- masive, de regulă de 4 ori mai grele decât o mașină unealtă convențională;
- motoare de acționare puternice cu posibilități de așchiere rapidă (în concordanță cu sculele moderne). Puterea și turația sunt de 4 ori mai mari (decât cu m-u convenționale);
- schimbător automat de scule cu capacitatea de la 8 la sute de scule și sistem de paletizare.

## Magazin de scule.(exemplu)





# Sistemul de paletizare:



**Precizie ridicată, de regulă 0,01- 0,001 mm.**

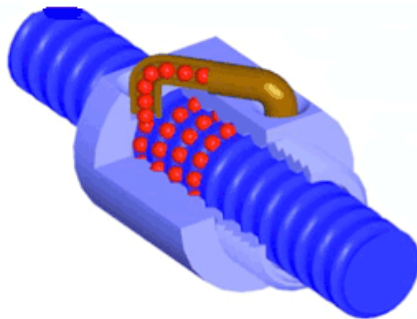


## Sistem de paletizare



# Cum se realizează precizia (1):

- şuruburi cu bile:

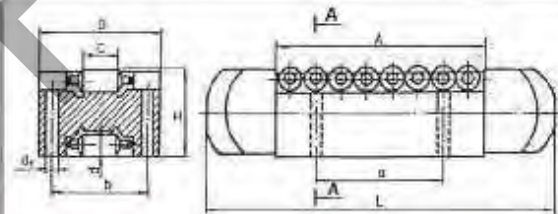
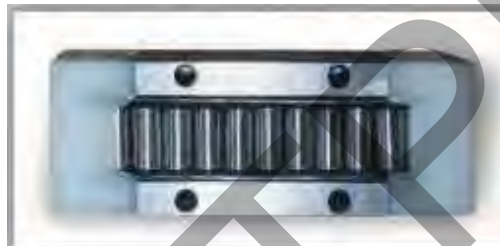
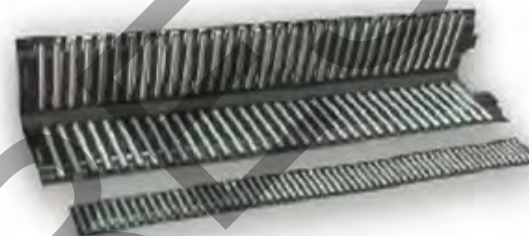
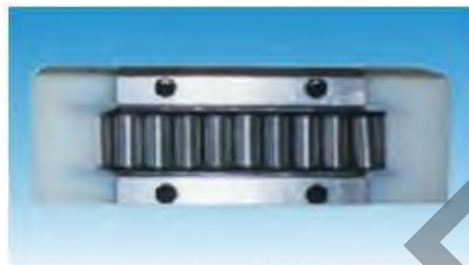


# Cum se realizează precizia (2):

- ghidaje de tip tanchete (elimină stick-slip-ul)

Recirculabile

Fixe

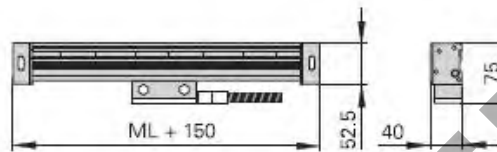
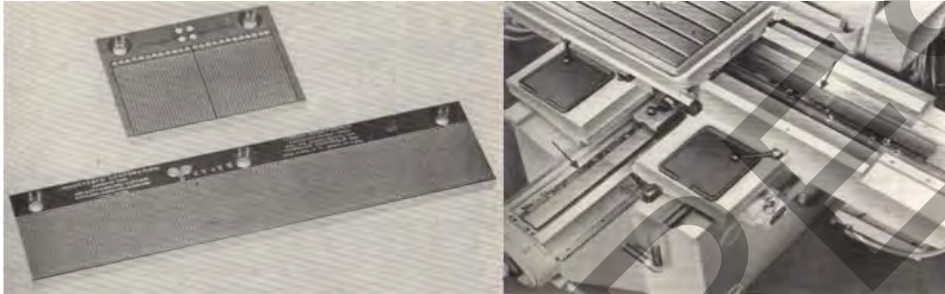


Tip		Abateri de la paralelism a ghidajului în stare montată <i>Deviation of the guideway from parallelism mounted</i> Parallelitätsabweichung der Führungsmontage			Viteza Speed Geschwindigkeit m/min
Type		Clasa de precizie <i>Class of accuracy</i> Genauigkeitsklasse			
Typ		1	2	3	
GRT 1	TRSU 1	1,5	2,5	3,5	10
GRT 2	TRSU 2	1,5	2,5	3,5	
GRT 3	TRSU 3	2	3	4	
GRT 4	TRSU 4	2	3	4	
GRT 5		3	4	5	
GRT 6		3	4	5	



# Cum se realizează precizia (3):

- traductare de deplasare.



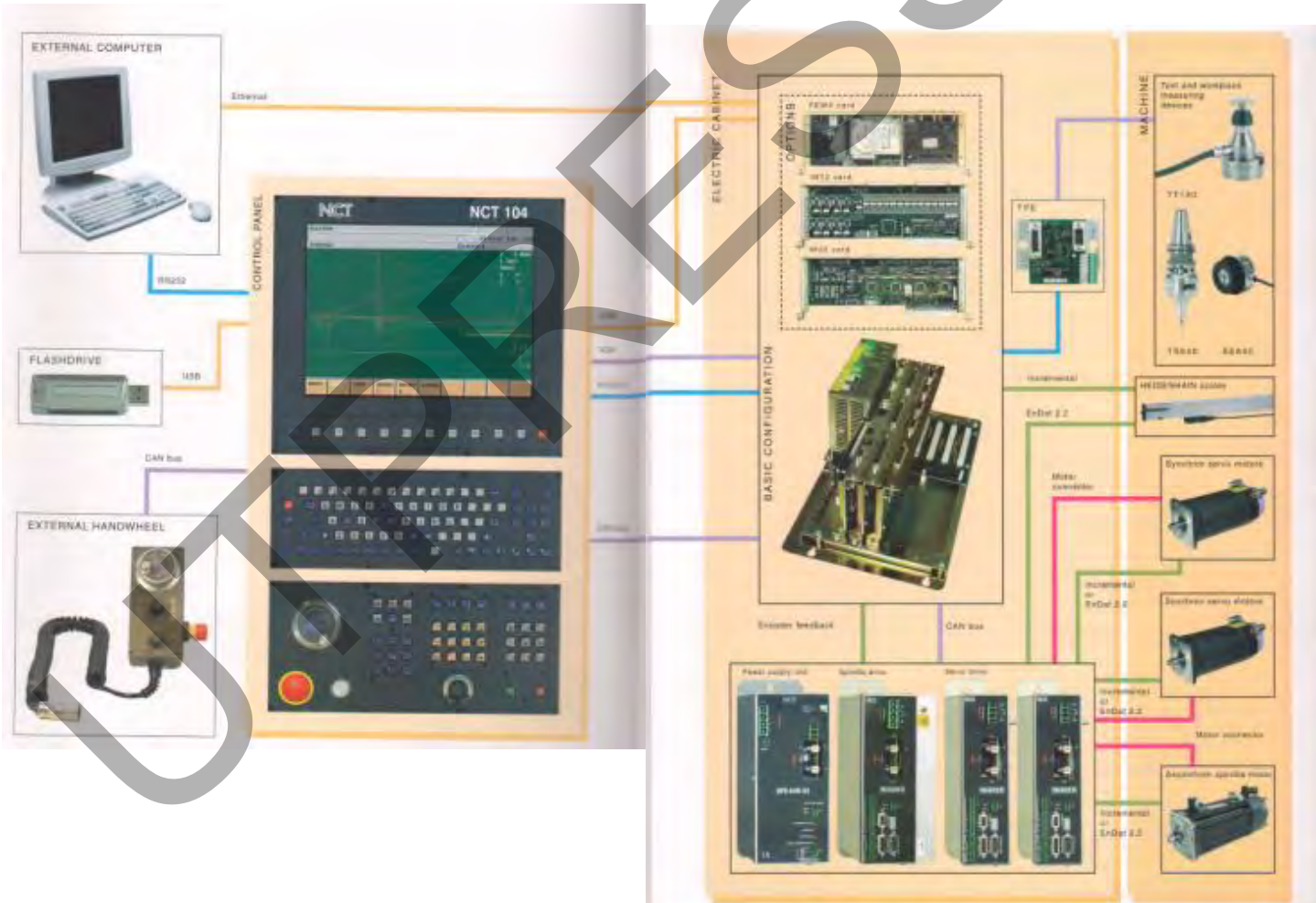
Heidenhain



Encoder



# Tendințe noi în gestionarea softurilor de comandă numerică



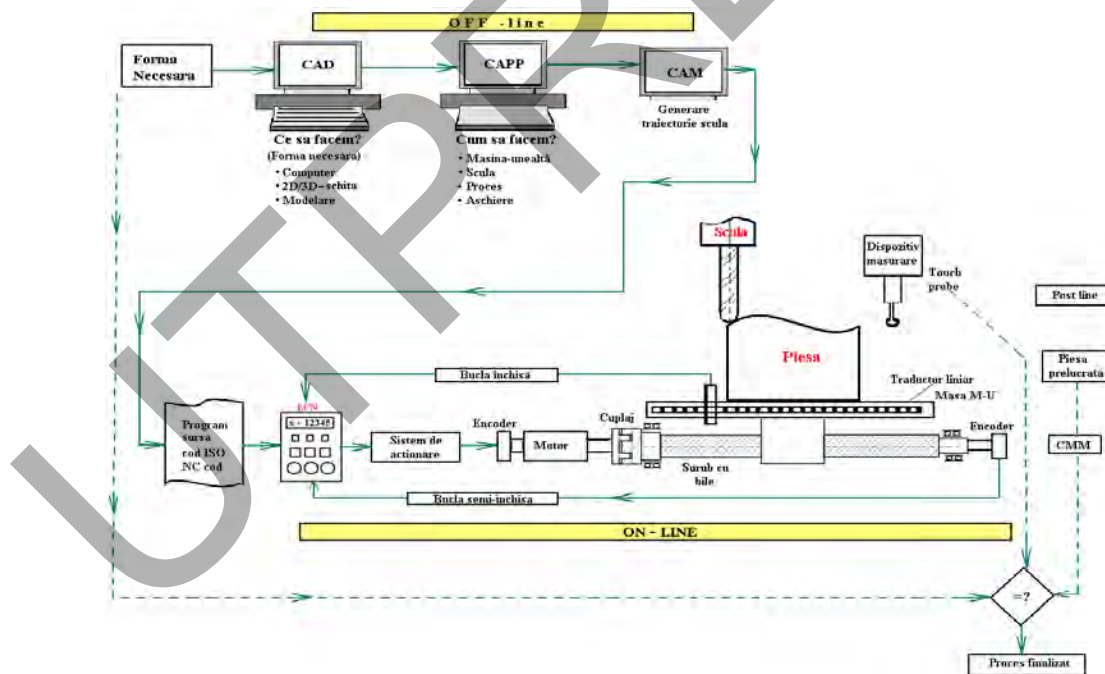
# Sistemul de prelucrare CNC

## CLASIFICAREA ACTIVITĂȚILOR

1. Off-line: CAD, CAPP, CAM

2. On-line: Prelucrare, monitorizare, măsurare pe mașină (touch probe)

3. Post-line: CAI (Computer Aided Inspection) prin CMM (Coordinate Measurement Machine)



# Aspecte generale privind programarea sistemelor numerice CNC

Sistemelor numerice CNC sunt alcătuite din două părți :

- ❖ Sistemul de comandă;
- ❖ Mașina-unealtă.

Activitățile desfășurate în cadrul unui sistem CNC sunt evidențiate în figura anterioară.

Suportul de curs abordează numai prima categorie de activități: generarea programului sursă de prelucrare.

Se evidențiază în principal două abordări:

*Programarea manuală*, specifică pieselor 2D, denumită și programarea ISO cod G;

*Programarea asistată de calculator*, specifică pieselor 3D, utilizând produse software specializate CAD, CAM.

Se evidențiază și alte sisteme de programare cum ar fi cel conversațional, WOP (Work shop oriented programming), etc.





# Programarea manuală

Suportul de curs este structurat , dpv al cunoștințelor, pe două niveluri:

Nivelul de bază, cunoștințele acumulate se concretizează în abilitatea de a întocmi programe de prelucrare simple;

Nivelul avansat, bazat pe programarea flexibilă (repetări de blocuri, cicluri fixe, subprogram, macrouri).

## Nivelul de bază

N10 G01 XB F400 ;

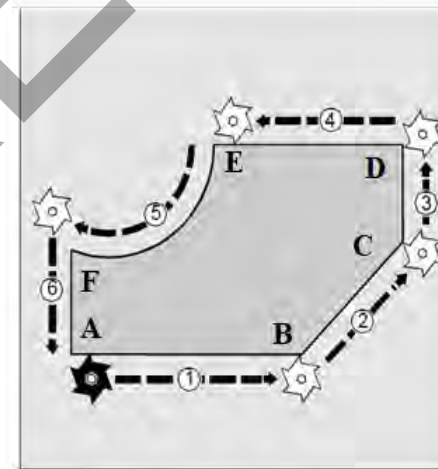
N20 XC YC ;

N30 YD ;

N40 XE ;

N50 G02 XF YF R ;

N60 G01 YA ;



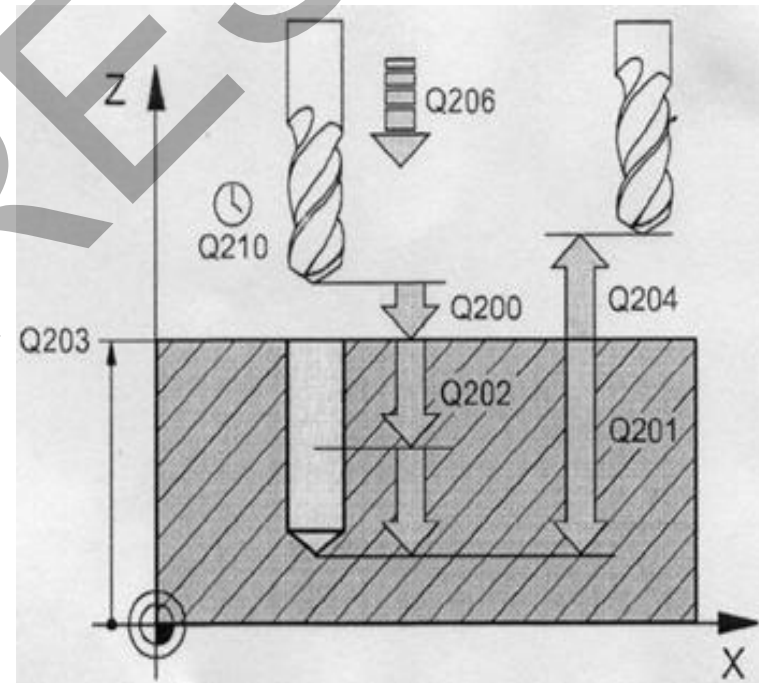
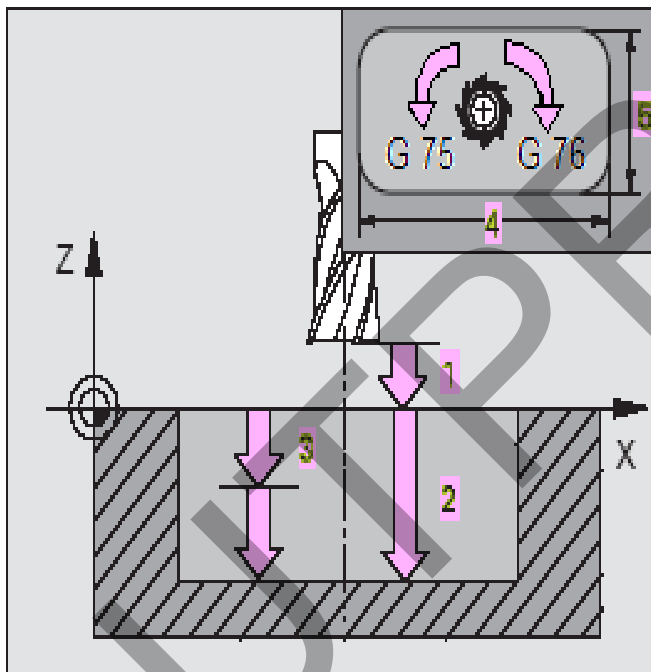
Programatorul trebuie să considere, pentru realizarea unui program corect, faptul că:

În programare se consideră că scula execută deplasarea;

Programarea fiecărei laturi a conturului se face în cadrul unui bloc.

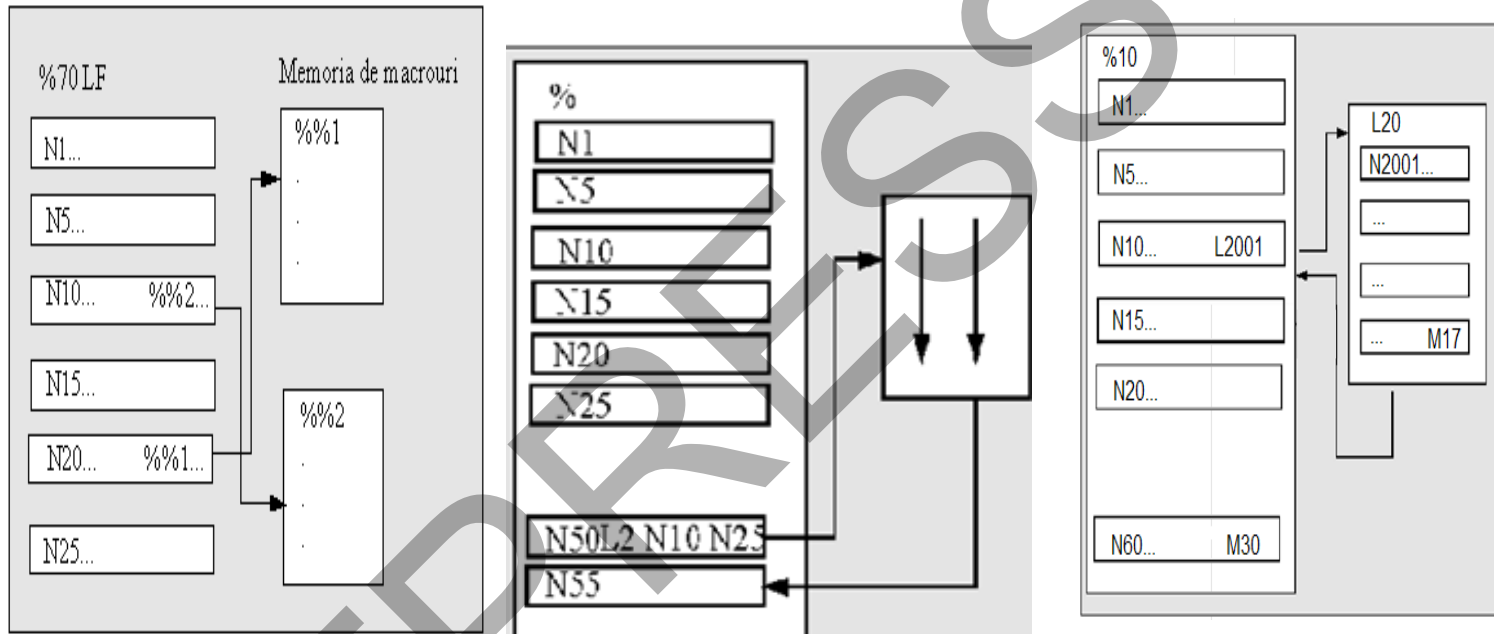
## Nivelul avansat-programare flexibilă

Repetări de blocuri Subprograme Macrouri Cicluri fixe



Cicluri fixe: Generarea automată a secvențelor de programare pentru prelucrări care presupun mișcări repetate

## Cap.1. Introducere



Repetările de blocuri conduc la reducerea activității de programare prin repetarea secvențelor de prelucrare identice.

Subprogramele, pot fi apelate în mod repetat în program, posibil cu parametri variabili cu scopul de a utiliza valori numerice variabile pentru parametrii.

Macrourele pot fi de tipul implementate în sistem la cererea beneficiarului.



## Cap.2. ARHITECTURA UNUI PROGRAM NC

Activitatea de codificare a informațiilor (elaborarea programului sursă NC) reprezintă numai o mică parte din activitatea de programare în ansamblu său.

Câteva din activitățile ce trebuie efectuate înainte de codificarea informațiilor sunt prezentate în continuare.

- **Analiza desenului piesei**

De regulă un desen finalizat în cadrul activității de proiectare poate fi utilizat în activitatea de programare cu anumite adăugiri.

În primul rând este vorba de alegerea punctului care va fi considerat originea piesei, Op. Funcție de locul ales pentru origine se trasează sistemul de coordonate.

În continuare, se analizează cotele și se evidențiază cotele lipsă conform sistemului de coordonate adoptat. După evidențierea acestora urmează calculul lor.



- **Stabilirea secvențelor de prelucrare**

Stabilirea secvențelor de prelucrare presupune în esență stabilirea tehnologiei de fabricație. Se va analiza tipul de prelucrare (degroșare, finisare), tipul de suprafață ce trebuie realizată, care este scula potrivită, în ce ordine se va realiza prelucrarea, etc.. Un aspect important, din punct de vedere a realizării programului, se referă la analiza tipului de structuri din alcătuirea piesei. Dacă sunt structuri care se repetă este avantajos să se apeleze în programare la tehnica subprogramelor. Poate anumite structuri se regăsesc și în alcătuirea altor piese deja prelucrate. În acest caz se pot prelua subrutine sau subprograme deja existente.

- **Dezvoltarea planului de prelucrare**

După stabilirea fazelor de prelucrare (secvențelor) este necesar ca pentru fiecare să se stabilească o succesiune de mișcări, de poziționare și cu avans de lucru, de tipul indicat în figura 2.3. Suplimentar se vor stabili punctele de schimbare a sculei, a paletelor (dacă există în dotarea sistemului numeric), curbele cele mai potrivite pentru apropierea / depărtarea sculei de contur, etc.

- **Întocmirea programului sursă de prelucrare**

Fiecare din pașii menționați anterior urmează a fi codificați în vederea constituirii blocurilor din programul NC. În acest scop sunt necesare, alături de informațiile privind tehnologia de prelucrare, și informații referitoare la limbajul ISO de programare, din punctul de vedere a adreselor implementate, a sintaxei limbajului.

Sintetizând aceste activități se pot evidenția următoarele proceduri (pași) de bază:

### 1. Pregătirea desenului piesei

- ❖ Definirea punctului zero al piesei;
- ❖ Indicarea sistemului de coordonate;
- ❖ Calculul eventualelor coordonate lipsă;

### 2. Precizarea secvențelor de prelucrare

- ❖ Care sculă se utilizează și pentru care contur?
- ❖ În ce ordine urmează să fie prelucrate elementele individuale ale piesei ?
- ❖ Care element individual se repetă (sau se rotește) pentru a fi memorat într-o subrutină
- ❖ Există în memoria de programe secțiuni de contur ale altor piese sau subrutine care ar putea fi utilizate la piesa în cauză?
- ❖ Există zero offseturi, rotiri, oglindiri, scalări necesare sau folositoare?

### 3. Pregătește un program de prelucrare

*Definește toate operațiile de prelucrare pas cu pas astfel:*

- ❖ Deplăsările cu avans rapid pentru poziționări;
- ❖ Schimbările de sculă;
- ❖ Definirea planului de prelucrare;
- ❖ Retragere pentru inspecție (echipamente CNC);
- ❖ Acționare arbore principal, lichid de așchiere pornit /oprit;
- ❖ Apelare date scule;
- ❖ Avansul;
- ❖ Corectarea traiectoriei;
- ❖ Aproximarea/retragerea de contur;

4. Compilarea (codificarea ) pașilor de prelucrare în limbaj de programare ( ISO, conversațional, etc.)

5. Asamblarea pașilor individuali în programul sursă de prelucrare.





## Structura de principiu a unui program sursă

Fiecare program trebuie să îndeplinească cel puțin două condiții esențiale: siguranță, cu prioritate maximă, urmată de ușurința utilizării. Deși programele sursă diferă unele față de altele se pot evidenția anumite structuri ce se repetă sub o formă sau alta, în fiecare program. În cazul centrelor de prelucrare prin frezare și strunjire, orice program este structurat pe patru secțiuni diferite:

1. Secțiunea de început program.  
prelucrare ;
2. Secțiunea pentru dezactivarea sculei “i”
3. Secțiunea pentru activarea sculei “i+1”  
prelucrare ;
4. Secțiunea de sfârșit de program



## Secțiunea de început program

%

N 0005 G91G28 X0 Y0 Z0; (deplasarea în punctul de referință)

N0010 T01 M06; ( schimbare sculă, activare scula T01)

N0015 G54 G90 S400 M3; (declarare zero piesă, programare în sistem absolut, pornire AP cu 400 rot/min )

N0020 G00 X10.0 Y15.0; (deplasarea în punctul de start)

N0025 G43 Z5.0 H01 (D01) M08; (poziționare pe Z, pornire lichid de așchiere);

N0030 G01 Y 65.558 F100; (conturare, viteză de avans 100 mm/min)

Urmează prelucrarea conturului – informații specifice fiecărei prelucrări



## Secțiunea pentru dezactivarea sculei T1

N0085 M09; (oprire lichid aşchiere )

N0090 G91 G28 G40 Z0 M 19; (revenire în punctul de referință, anulare corecție de lungime, oprire orientată AP )

N0095 M01 (stop opțional)

N0100 M05 (oprire AP)

N0105 T02 M06 (schimbare sculă, activare sculă T2)

## Secțiune pentru activarea sculei T2

N0110 G54 G90 S600 M3;

N0115 G00 X50.0 Y50.0;

N0120 G43 Z50.0 H02 (D02) M08;

Urmează prelucrarea cu scula T2

Deoarece prelucrarea găurii presupune și utilizarea sculei T3 (burghiul) în program urmează, în succesiune secțiunile doi și trei.



## Secțiunea de sfârșit de program

N0150 M09;

N0155 G91 G28 G40 Z0 M19;

N0160 G28 X0 Y0;

N0165 M30;

%

Unele echipamente pot avea structura diferitelor secțiuni ușor modificată.

Frecvent se recomandă ca în secțiunile de început de program și activarea sculei "i+1" să fie programate anumite condiții inițiale cum ar fi: anulare corecții, anulare cicluri etc.



# Cap.3. PROGRAMAREA MANUALĂ CNC

## Aspecte generale

Controlul mașinii-unelte este realizat de către programator care redactează instrucțiuni pentru echipamentul CNC prin care specifică:

- Care scula urmează a fi încărcată în arborele principal al mașinii
- Care sunt condițiile de așchiere (avans, turații, lichid de așchiere ON/OFF)
- Punctul de început și sfârșit pe fiecare segment
- Cum se deplasează scula în raport de mașină
- ✓ punct cu punct specific pentru mașini de găurit
- ✓ conturare specific pentru mașini de frezat, strunuri, etc.

Traietoriile sunt alcătuite din segmente de tip linear sau arce de cerc.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

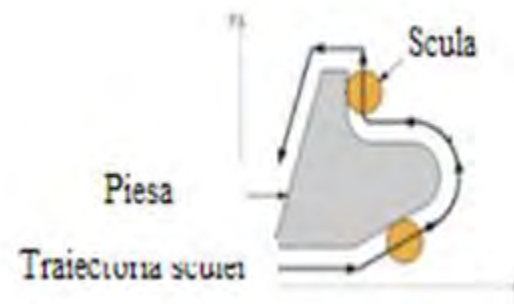
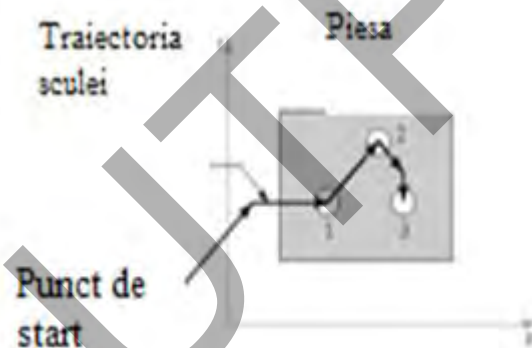
# Controlul deplasării sculei

## Sisteme punct cu punct

- Denumire și sisteme de poziționare
- Scula se deplasează prin aer la o anumită poziție unde execută o operație (ex. burghiere)
- Se utilizează și în comanda roboților

## Sisteme cu controlul continuu al deplasării

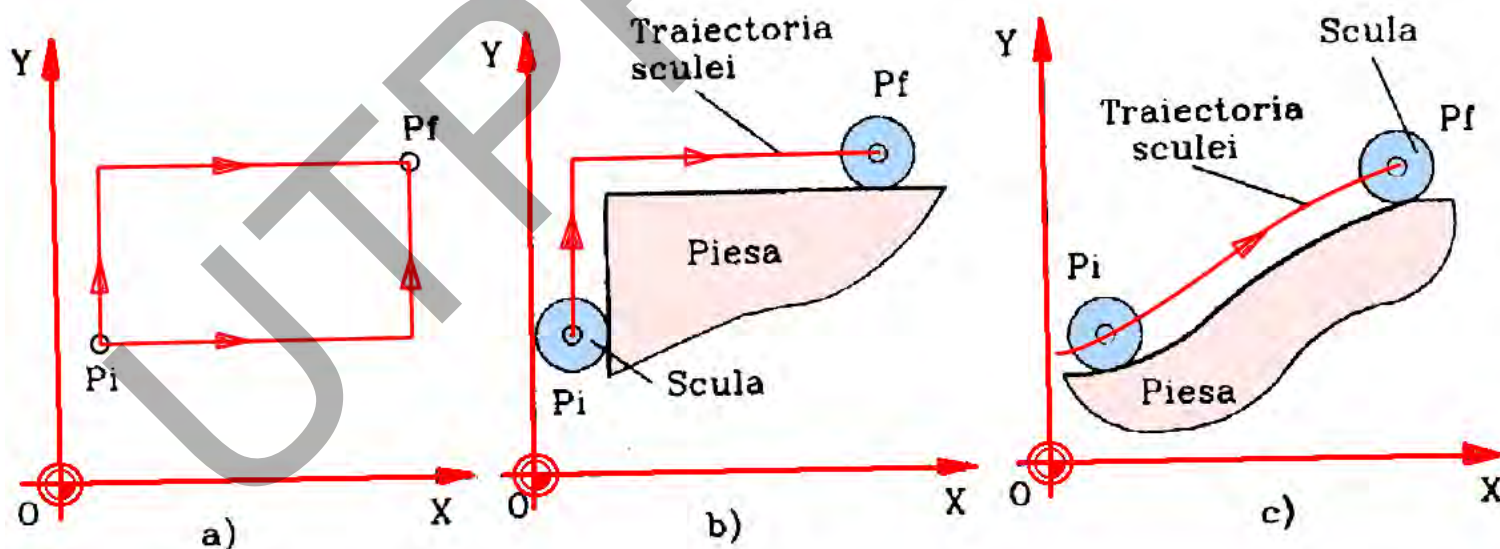
- Se numesc și steme de conturare
- Pe durata deplasării scula așchiază



# Clasificarea ECN

- posibilități de prelucrare: PCP, PL, C;
- programarea cotelor: absolut (G90), incremental (G91);
- circuite de comandă: bucla închisă, deschisă;
- realizare: hardware, software.

## Posibilități de prelucrare:



## Notă

- ***Se consideră că scula se deplasează în raport de piesă indiferent de situația reală.***
- Poziția sculei este descrisă în sistemul de coordonate (cartezian sau polar). Nu este restrictiv.
- **Echipamentele CNC cer programarea în formatul G cod.**
- G-cod este un limbaj de programare NC de tipul “low-level.”
- Codurile G implementate sunt specifice pentru fiecare mașină - se utilizează postprocesoare.
- **Limbajul descrie deplasarea sculei prin comenzi simple.**



# Sintaxa limbajului de programare

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## Număr secvență (Identificator **N**)

Identifică nr. Blocului, în ordine crescătoare fără a fi necesară o secvențiere continuă

## Funcția pregătitoare (Identificator **G**)

Pregătește mașina-unealtă / echipamentul pentru o anumită operație

## Informații geometrice (Identificator **X, Y, Z, A, B, C**)

Descriu locația sculei, orientarea axelor, pentru deplasarea sculei

## Funcții tehnologice (Identificator pentru avans **F**, Identificator pentru turație **S**)

Descriu parametrii regimului de așchiere

## Funcția sculă (Identificator **T**)

Specifică scula ce urmează a fi utilizată și offseturile acesteia

## Funcții auxiliare-miscellaneous (Identificator **M**)

Specifică un anumit mod de operare-comutații

## Sfârșit de bloc (Identificator **EOB, ;**)



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

# Structura blocului

Cuvânt	Format	Semnificație
N	4 digiti, int	Număr bloc
G	2 digiti, int	Funcție pregătitoare
M	2 digiti, int	Funcție auxiliară
X,Y,Z,A,B... W,I,J,K	Numere reale	Poziția pe axe, unitate de măsură
F	4 digiti, int	Valoarea vitezei de avans, mm/min
S	4 digiti, int	Valoarea turației, RPM
T	2/4 digiti, int	Număr sculă/registru de corecție



# Poziția sculei în zona de lucru

Cap.1.  
Introducere

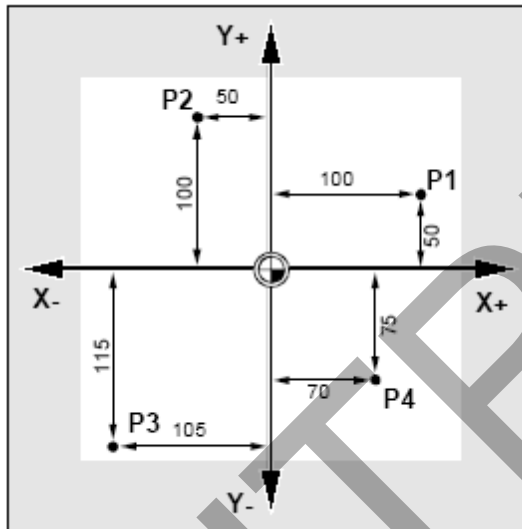
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## Sistemul de coordonate CARTEZIAN

Poziția punctului exprimată prin 2/3 coordonate

Freza

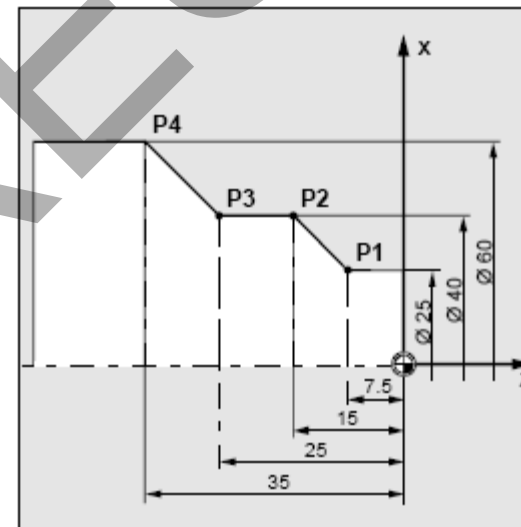


$P1(x=100, y=50)$

.....

$P3(x=100, y=-115)$

Strung



$P1(x=25, z=-7.5)$

.....

$P4(x=60, z=-35)$



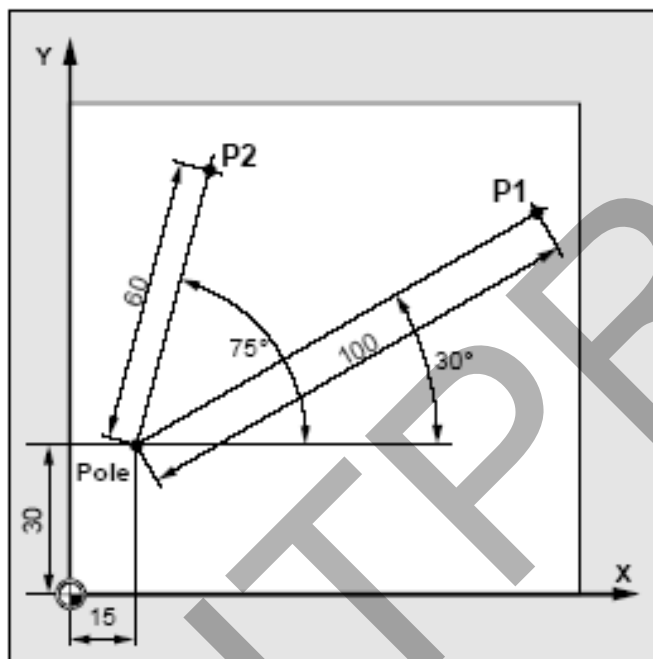
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

# Sistemul de coordonate POLAR

Poziția punctului exprimată prin rază și unghi



$P1(R=100, \alpha=30^\circ)$

$P2(R=60, \alpha=75^\circ, \text{ sau } \alpha=45^\circ)$

Poziția polului se precizează diferit funcție de echipament. De asemenea, modul de programare este specific fiecărui echipament.



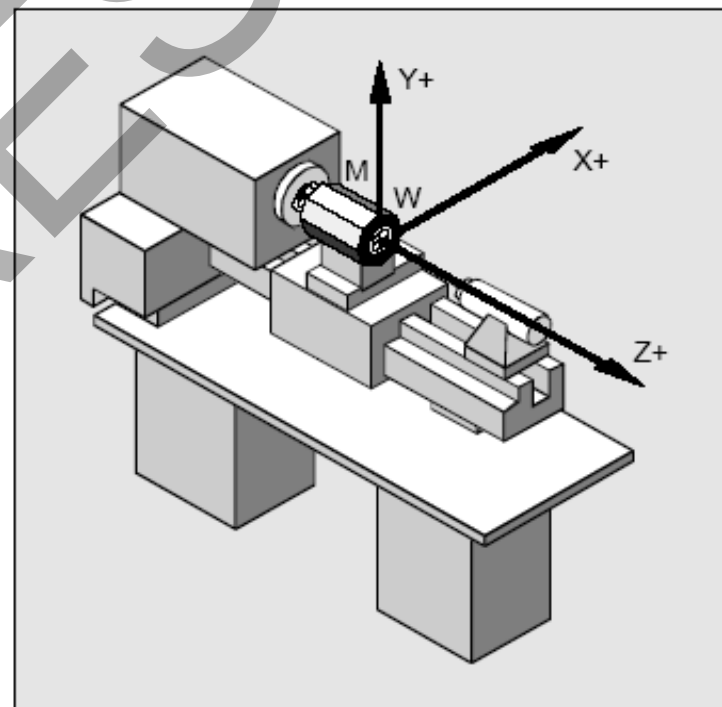
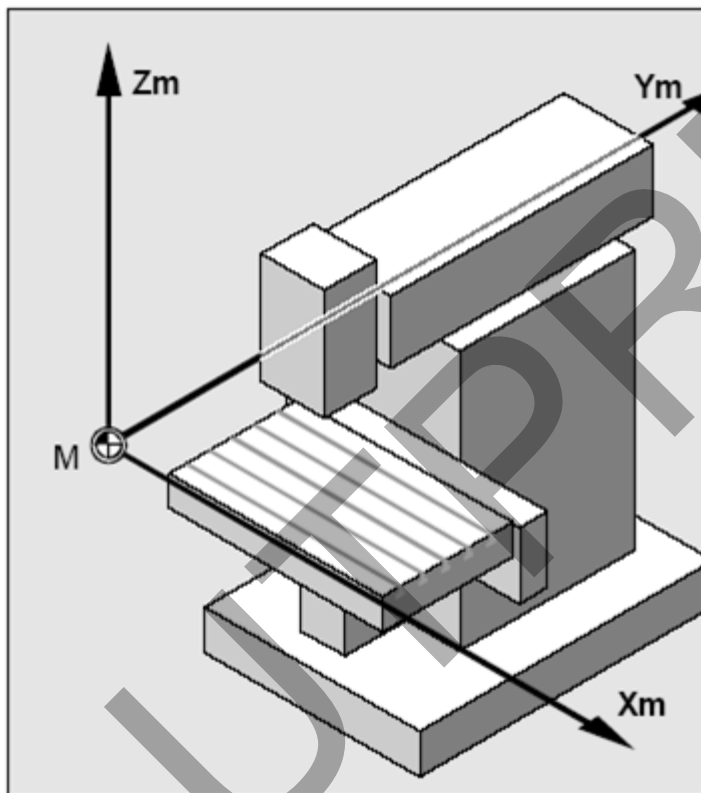
# Localizare sistem de axe

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## Sistemul de axe al mașinii



Originea fixă stabilită de constructor hard / soft



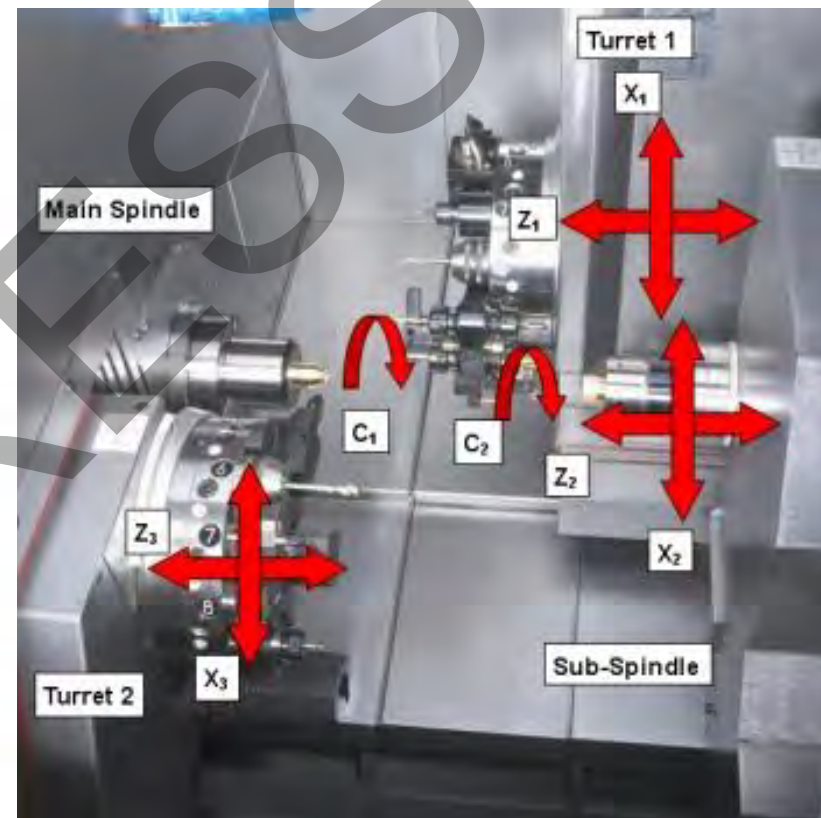
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO



Sisteme de axe  
Centru de prelucrare



Strung



## Machine zero M



The machine zero M is defined by the manufacturer and cannot be changed. When milling, it is in the origin of the machine coordinate system, and when turning, on the contact surface of the spindle nose.

## Workpiece zero W



The workpiece zero W, also referred to as the program zero, is the origin of the workpiece coordinate system. It can also be freely selected and when milling, it should also be located at a position in the drawing from which most of the dimensions are measured.

When turning, the workpiece zero is always located on the rotary axis and mostly on the plane surface.

## The reference point R



The reference point R is approached when setting the measuring system, since in most cases the machine zero cannot be approached. The control system will thus find its reference point in the position measuring system.

## The toolholder reference point T



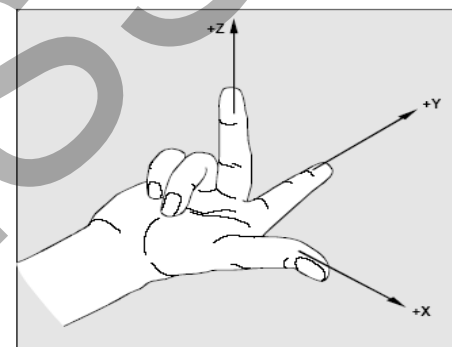
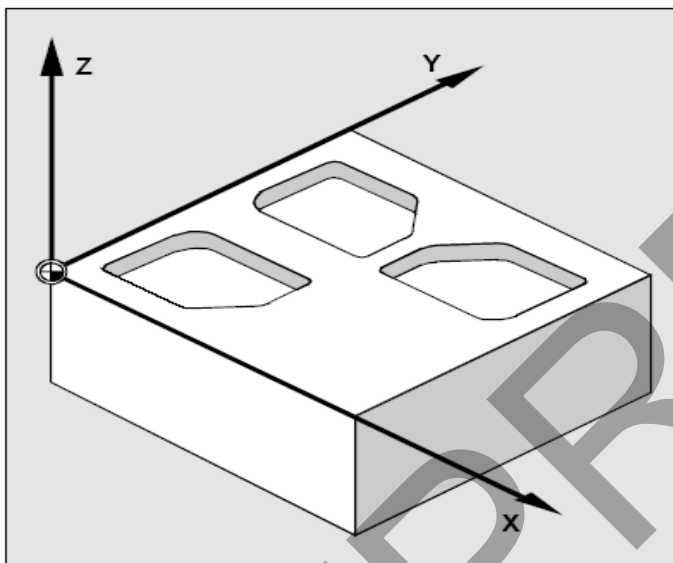
The toolholder reference point T is important for setting up with default tools. The lengths L and Q shown in the diagram below are used as tool calculation values and are entered in the tool memory of the control system.

# Sistemul de axe al piesei

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO



Regula mâinii drepte

Alegerea originii piesei este dictată de:

- ❖ Desenul piesei;
- ❖ Posibilitatea de atingere a punctului;
- ❖ Dacă nu este impus, se recomandă colțul din stânga față al piesei.

Se pot declara mai multe origini în cadrul unui program.





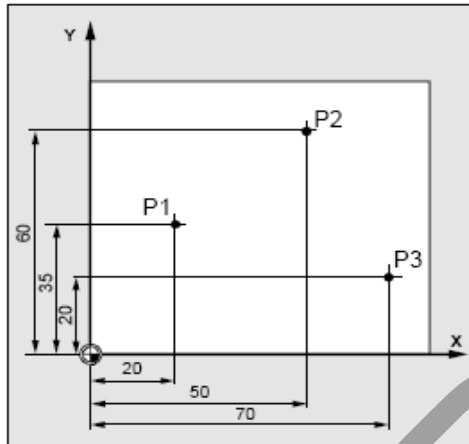
# Cotare

Cap.1.  
Introducere

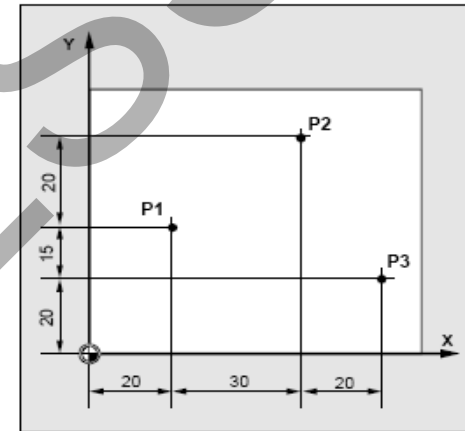
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

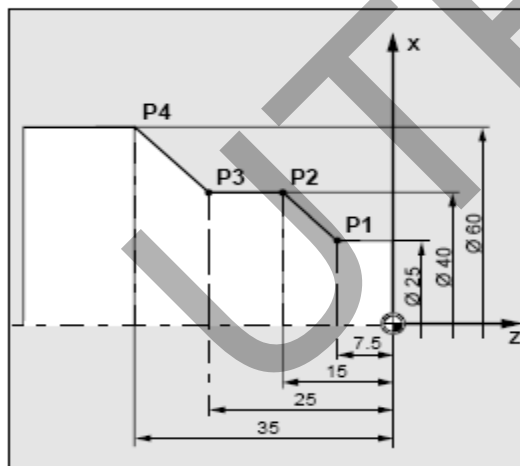
## Absolută



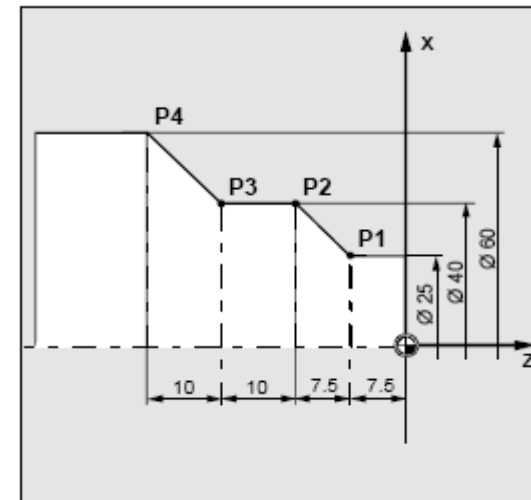
## Incrementală



## Frezare



## Strunjire

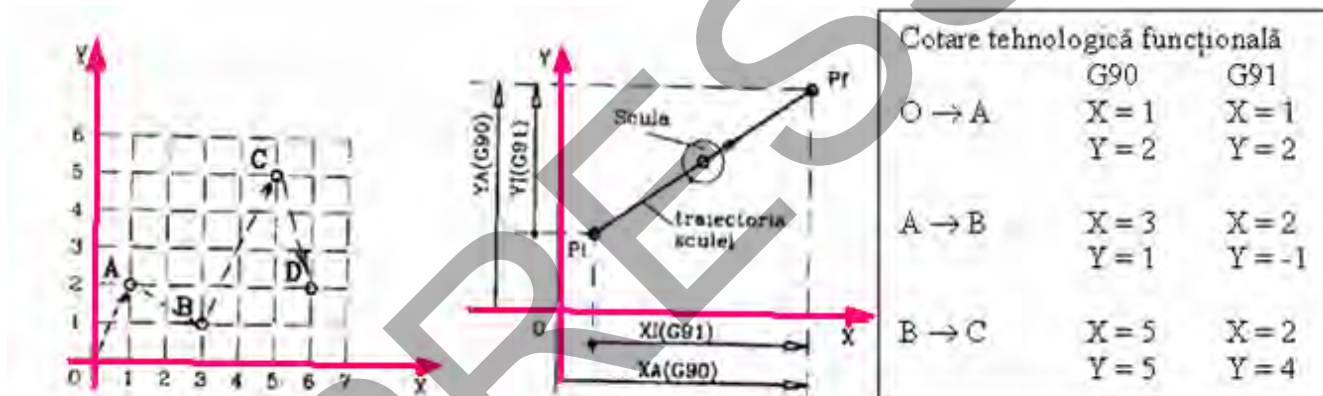


Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

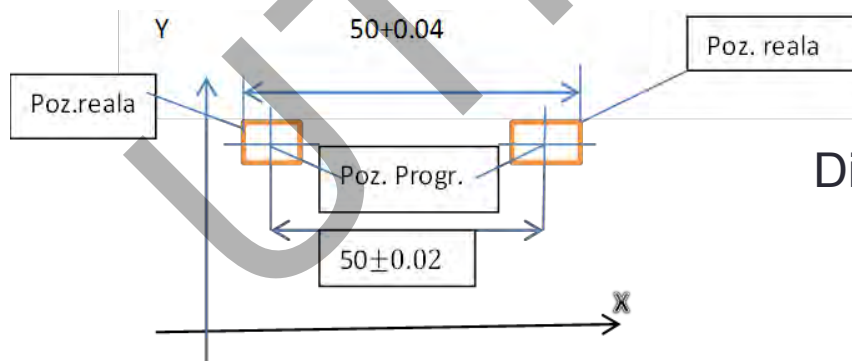
Cap.3.  
Limbaj ISO

# Programarea în sistem absolut/incremental



În ce poziție trebuie să deplasez scula?  
(G90)

Cât de departe trebuie să fie scula?  
(G91)



Discuție: G90/G91  
Închidere lanț dimensiuni



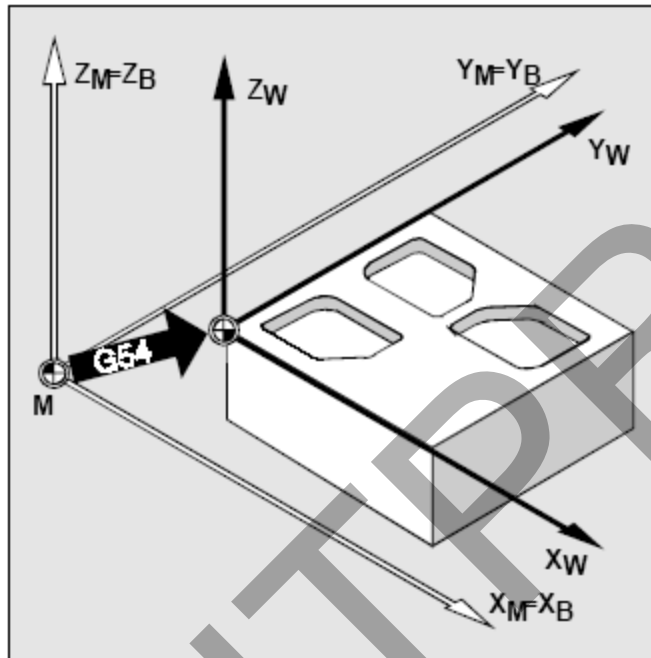
# Declararea originii piesei

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

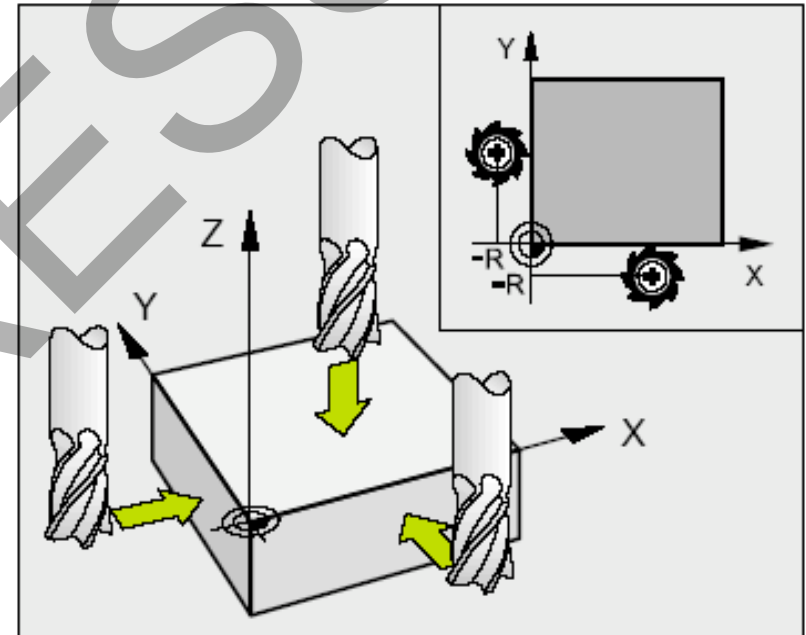
Cap.3.  
Limbaj ISO

Prin programare



Valorile deplasării sculei din originea mașinii în originea piesei se introduc pe panoul echipamentului, în regimul OFFSET SETTING –WORK SETTING

Prin alegere



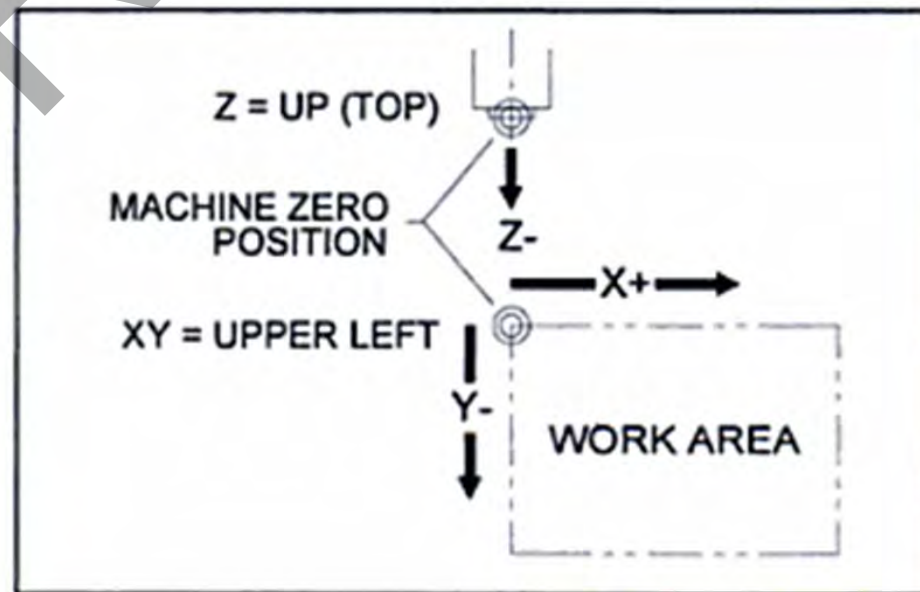
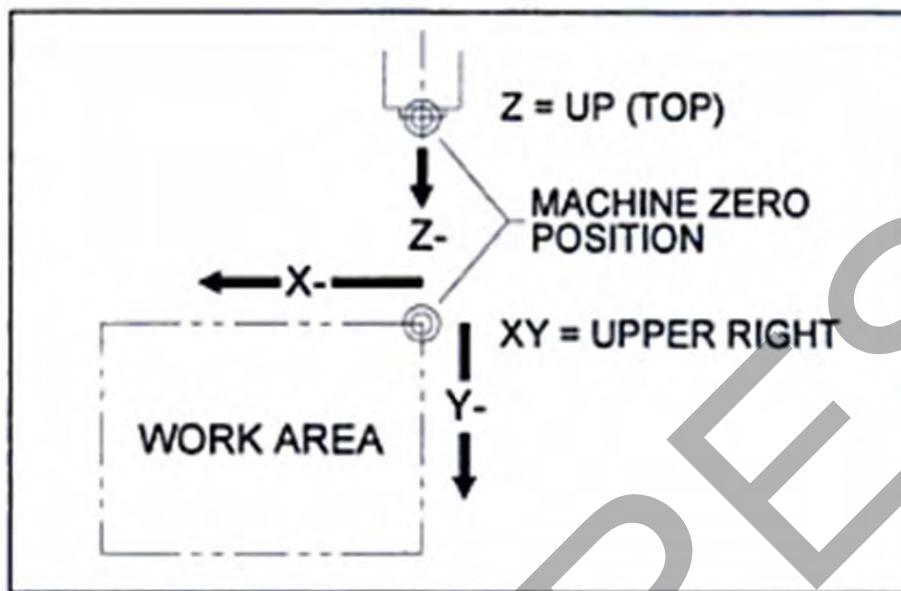
Se poziționează scula în origine și se programează: `G92 X0 Y0, Z0`



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO



Cap.1.  
Introducere

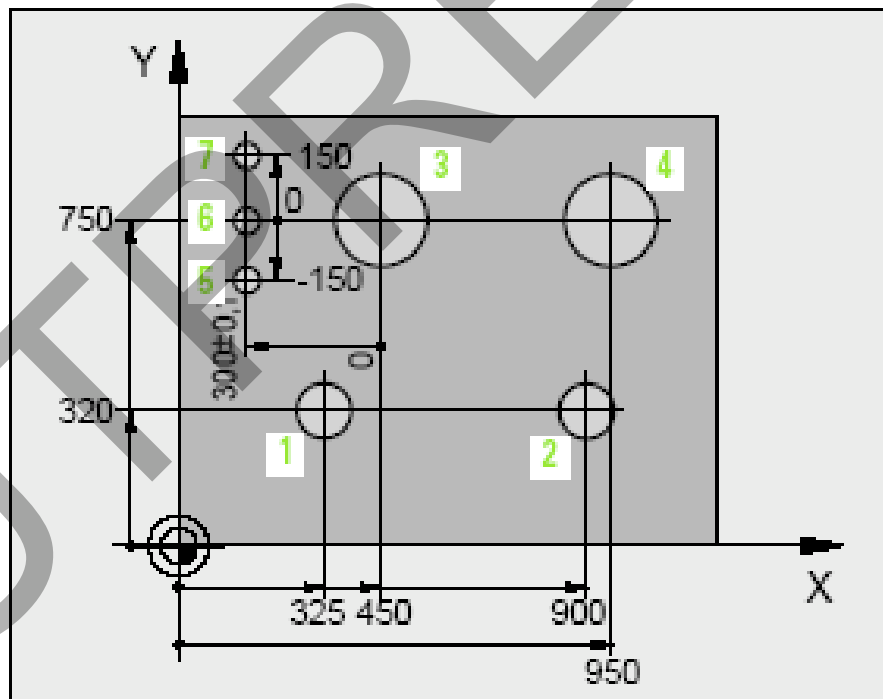
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

# Utilizarea a două origini

Punctele 1,2,3,4 pot fi programate în raport cu originea sistemului de coordonate al piesei.

Punctele 5,6,7 sunt cotate în raport de punctul 3. Este convenabilă alegerea acestuia ca origine suplimentară. Facilitatea aceasta se programează punctual pentru fiecare echipament CNC.



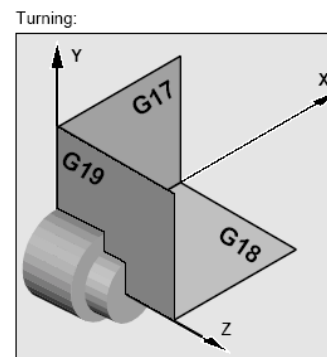
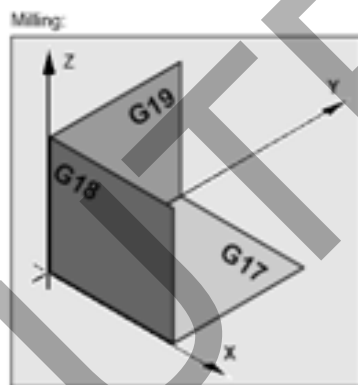
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

# Selectare plan de prelucrare

Definește planul în care are loc prelucrarea (2D). A treia axă, perpendiculară pe plan, este de poziționare. Sistemul se numește 2 ½ axe



## Limbajul de programare ISO (G cod)

Limbajul de programare ISO, singura alternativă de întocmire a programului sursă (singurul recunoscut de echipament) până la sfârșitul anilor '80.

Se caracterizează prin utilizarea unor coduri de tip G și M alături de codurile pentru programarea sculei, a condițiilor de așchiere și evident a deplasării sculei.

Programarea se face respectând standardul RS274 NGC. Există diferențe între limbajele ISO implementate pe echipamente CNC (NC).

În cele ce urmează vor fi evidențiate entități care se regăsesc la majoritatea echipamentelor aflate în exploatare azi.

Limbajul numeric de programare este înainte de toate un limbaj informatic evoluat adică posedă un vocabular și o sintaxă.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## Principalele aspecte ce trebuie evidențiate (studiate) se referă la:

- ☀ Caracterele utilizate;
- ☀ Numere utilizate;
- ☀ Variabilele (parametrii);
- ☀ Cuvintele;
- ☀ Instrucțiunile;
- ☀ Blocurile de instrucțiuni;
- ☀ Secvențele de blocuri de instrucțiuni;
- ☀ Structurile de control;
- ☀ Subprogramele rezidente și concepute.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

# Caractere utilizate

- Cifrele de la 0 la 9
- Literele: majusculele alfabetului latin A, B, C, etc.
- Semne și operatori matematici +, - <> ≤ ≥ =
- Caractere particulare
  - Început program %
  - Început comentariu (
  - Sfârșit comentariu )
  - Sfârșit de bloc LF, ;
- Caractere inerente, recunoscute de sistem, fără a avea însă un anumit efect.

# Numere utilizate

Întregi

Reale

Utilizate ca:

- Valori afectate variabilelor
- Argumente pentru unele funcții
- Constante în diferite expresii aritmetice

Se folosesc numerele în baza 10. Se pot utiliza și în baza 2 (precedate de %) și hexazecimal (precedate de \$) la unele CNC-uri

# Variabile - parametrii

Sunt definite apriori:

- Numărul de variabile utilizate
- Tipul de informație pe care îl reprezintă
- Identificatorul acestor variabile

## Variabile programate

- Utilizează diferiți identificatori predefiniți;
- Valorile acoperă un domeniu larg;
- Afectarea unei valori pentru o variabilă se face prin “=”

## Parametrii externi

- Parametrii rezervați pentru comunicarea între CNC și automat;
- Parametrii utilizați de CNC, accesibili programatorului pentru citire;
- Parametrii utilizați de CNC, accesibili programatorului pentru scriere



# Cuvintele limbajului

Cel mai mic ansamblu de caractere care posedă o semnificație independentă de alte caractere care pot urma și de care sunt despărțite prin operatori, semne de punctuație, etc.

## Cuvinte alcătuite dintr-o singură literă:

- X, Y, Z:** sistemul primar al axelor de coordonate liniare;
- A, B, C:** axe de rotație;
- U, V, W:** sistemul secundar al axelor de coordonate;
- I, J K:** coordonatele centrului unui cerc, parametrii de interpolare;
- P, Q:** utilizate în cicluri;
- R:** raza cercului, cuvânt utilizat în cicluri;
- F:** funcție ce definește viteza de avans;
- S:** funcție ce definește viteza de așchiere;
- T:** funcție ce definește scula.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

H, D: funcție ce definește registrii de corecție;  
L: funcție ce definește un subprogram ;  
N: eticheta pentru identificarea unui bloc.

Cuvinte alcătuite dintr-o literă urmată de un număr:

Gx, Gxx, Gxxx: funcții pentru modificarea stării logice a echipamentului, funcții pregătitoare.

Mx, Mxx, Mxxx: funcții de apelare a unui subprogram, de modificare a logicii ECN, funcții auxiliare.

# Instrucțiunea

Reprezintă entitatea cea mai mică, alcătuită din cuvinte și numere, care are o semnificație pentru modificarea fie a stării fizice a mașinii, fie a celei logice a echipamentului.

## Categoriile de instrucțiuni:

- ❖ Pentru deplasarea pe axele mașinii
- ❖ Relative la intervenții și mesaje adresate operatorului
- ❖ Pentru structurarea programului
- ❖ Referitoare la subprogramele rezidente

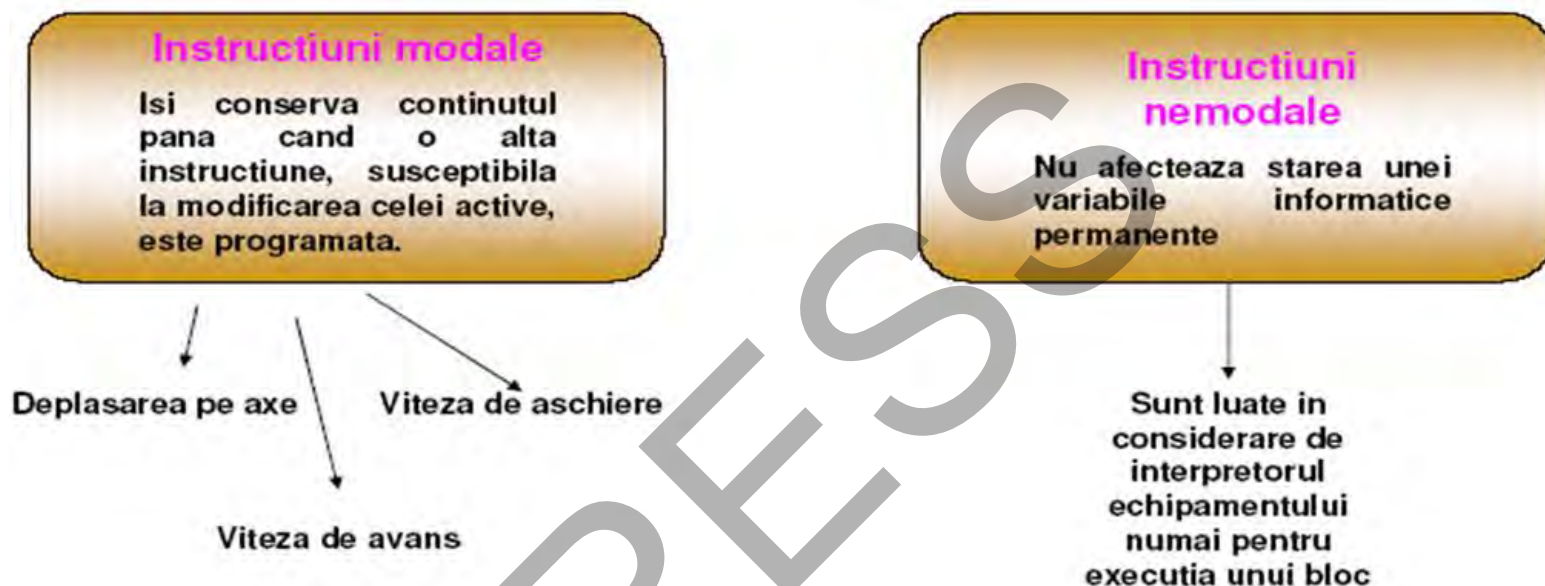
## Moduri de constituire:

- ❖ Un cuvânt unic
- ❖ Un cuvânt urmat de un număr
- ❖ Un ansamblu de cuvinte urmate sau nu de un număr

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO



```
N20 G01 X20.8 Y54.6 F89;
N25 X45;
N30 Y-34.6 ;
N35 G00 Z120;
N40 X0 Y0;
```

Instrucțiunea G01 din blocul N20 este activă până în blocul N35

```
N20 G01 X.....Y.....F.....;
N30 G00 Z150;
N35 T02 M06;
N40 G81 Z-50 R5. H2 F50.
```

Instrucțiunea M06 (schimbare sculă) este activă numai în blocul N35

# Bloc de instrucțiuni

Un ansamblu de instrucțiuni terminate cu caracterul care definește sfârșit de bloc (LF,;)

Este identificat prin intermediul etichetei reprezentată printr-un anumit cuvânt, urmată de un grup de maxim 4-5 cifre

Utilizarea lui face ca un bloc să fie considerat de programator ca o linie de program.

Deși nu toate liniile din program respectă riguros definiția blocului, se obișnuiește să fie toate numite blocuri.

Execuția blocului este aceeași, indiferent de ordinea cuvintelor în bloc.

Câteva aspecte generale se impun a fi prezentate la redactarea unui program „NC” pentru EMC (Enhance Machine Control):

- ❖ Sunt admise spații oriunde în cadrul blocului. Rezultatul interpretării blocului este același chiar dacă lipsesc anumite spații „normale”:

`G0X + 0.123Y7 ≡ G0 X+0.1237Y7;`

- ❖ Se pot utiliza atât majuscule cât și literele mici;

- ❖ Liniile goale sunt ignorate.

Există câteva limitări privitoare la structura blocului specifice fiecărui echipament.





# Secvența de instrucțiuni

Un ansamblu de blocuri consecutive identificat prin două etichete (de exemplu Nxxx până la Nyyy), prima afectată primului bloc al secvenței, iar cea de-a doua ultimului.

		N120	
N120	G00 X... Y...;		G00 X... Y...;
	G01 X... Y...F..;		G01 X... Y...F..;
	G02 X... Y... R...;		G02 X... Y... R...;
	G0 Z...;		G0 Z...;
N150	L5=L5+1		L5=L5+1
		N150	

Programarea este specifică fiecărui tip de echipament.  
Astfel, la

➤ **CNC de tip DIALOG 4, 10:**

Nxxx L=m Nyyy Nzxx

N30 L1 N20 N60

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## ❖ CNC de tip FANUC Series Oi-MC

Oferă mai multe posibilități privind utilizarea subprogramelor.

O □□□□ nume subprogram

.  
M99 sfârșit subprogram

Apelare subprogram

M98 P O O O O □□□□ numele subprogramului  
Număr de repetări

În programul principal

O 0001;

N0010...;

N0020...;

N0030 M98 P1010;

N0040...;

N0050...;

N0060...;

O1010

N1005...;

N1010...;

N1015...;

...

N1045 M99 P0050;

# Sucesiunea cuvintelor în bloc

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

N \_ \_ G \_ \_ X \_ \_ Y \_ \_ Z \_ \_ F \_ \_ S \_ \_ T \_ \_ D \_ \_ M \_ \_ H \_ \_ ;

Sucesiunea de execuție – logica bunului simț

**Bloc principal:** conține toate cuvintele necesare derulării unei secvențe de prelucrare din programul sursă. De regulă, se declară bloc principal acela în care se apelează o sculă nouă.

Se marchează cu: `25 G00.....`

**Bloc:** conține cuvinte prin care se asigură continuarea secvenței de prelucrare

Se marchează cu `N25 G00 ...`

**Bloc opțional:** un bloc, principal sau nu, care poate să nu fie executat pe parcursul derulării programului

Se marchează: `/N20 , /:20`

# Structuri de control

Un ansamblu de instrucțiuni care permit derularea unui program și altfel decât “linear”:

Aceste structuri permit:

- ❖ Identificarea programelor, subprogramelor;
- ❖ Apelarea subprogramelor și a secvențelor de program;
- ❖ Salt la diferite etichete;
- ❖ Temporizări;
- ❖ Marcarea sfârșitului de program principal și subprogram.

## Exemple

### Salt condiționat / necondiționat.

Prin această facilitate se întrerupe derularea “liniară” a programului sursă. Este echivalentul instrucțiunii GOTO, JUMP etc. din diferite limbaje informatice. Modul de programare este foarte diferit.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## Salt necondiționat

### Echipamentul FANUC Series Oi-MC

Echipamentul poate controla desfășurarea fluxului informațional prin utilizarea expresiei

GØTØ GØTØ n;n-numărul blocului(1...99999)

GØTØ 1;

GØTØ #10;

## Salt condiționat

IF [expresie condițională] GØTØ n

IF [#1 GT 10] GØTØ 2

Dacă cond nu este satisfăcută

→ Procesare

N2 G00 G91 X100 ←

Dacă cond este satisfăcută

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

**Mod de apelare:** diferă foarte mult de la echipament la echipament

### ❖ Temporizarea

Este asociată unei condiții de continuare a programului până când variabila nu atinge valoarea zero. Instrucțiunea de temporizare este:

G04 adresă xxx

Sunt utilizate diferite adrese în asociere cu G04: X, F, etc. Cifra indică valoarea temporizării în s.

G04 X 2.2 , temporizare de 2.2 s.

## Subprograme

Există două categorii distincte:

- ❖ Rezidente în memoria echipamentului
- ❖ Redactate de utilizator

Reluarea programului de prelucrare necesită marcarea sfârșitului subprogramului.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

# Structura de principiu a unui program sursă

**Pas 1:** Selectare program

Intrări: Nume program

Unitate de măsură în program

**Pas 2:** Apelare date sculă

Intrări: Nume program

Unitate de măsură în program

**Pas 3:** Schimbare sculă

Intrări: Coordonatele punctului de schimbare a sculei

Corecția sculei

Separate: Avansul (poziționare rapidă)

Funcții de tip M

**Pas 4:** Apropierea de poziția de start

Intrări: Coordonatele punctului de start

Anulare corecție de rază

Separat: Avansul (poziționare rapidă)

Funcții de tip M



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

**Pas 5:** Deplasare la adâncimea de așchiere

Intrări: Coordonatele planului de așchiere

Avansul (poziționare rapidă)

**Pas 6:** Deplasarea la primul punct de pe contur

Intrări: Coordonatele punctului

Activare corecție de rază

Separat: Viteza de avans în prelucrare

**Pas 7:** Prelucrarea până la primul punct de pe contur

Intrări: Toate datele necesare parcurgerii tuturor elementelor de contur

**Pas 8:** Deplasare la poziția de stop

Intrări: Coordonatele punctului final

Anulare corecție de rază

Separat: Funcții M (oprire arbore principal)

**Pas 9:** Retragerere sculă

Intrări: Coordonatele punctului din planul de siguranță

Separat: Funcții M (sfârșit de program)

**Pas 10:** Sfârșit program



# Programul sursă

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## DEZVOLTAREA PROGRAMELOR SURSA

- Informatii privitoare la pozitie
- Programarea comenzilor pentru deplasare
- Programarea datelor referitoare la scula

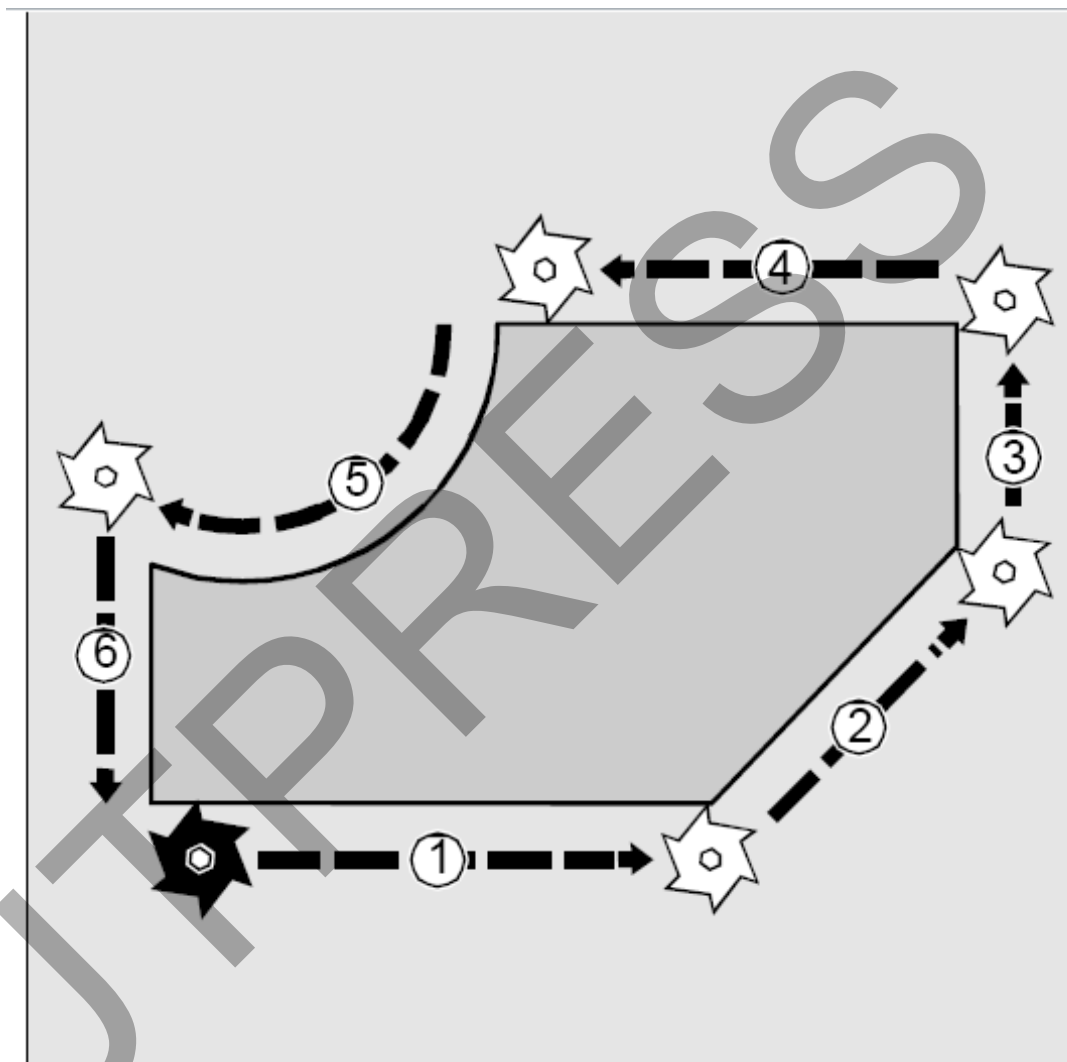
- Primele doua caractere de tip litera sau caracter special litera
- Pana la 24 de caractere

%	Nume program		comentariu
BLOC	CUVÂNT	CUVÂNT..;	comentariu
N10	G00	X20;	comentariu
N100	M30	;	Sfârșit program

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO



**Important!** Evidențierea curbelor care definesc conturul piesei.

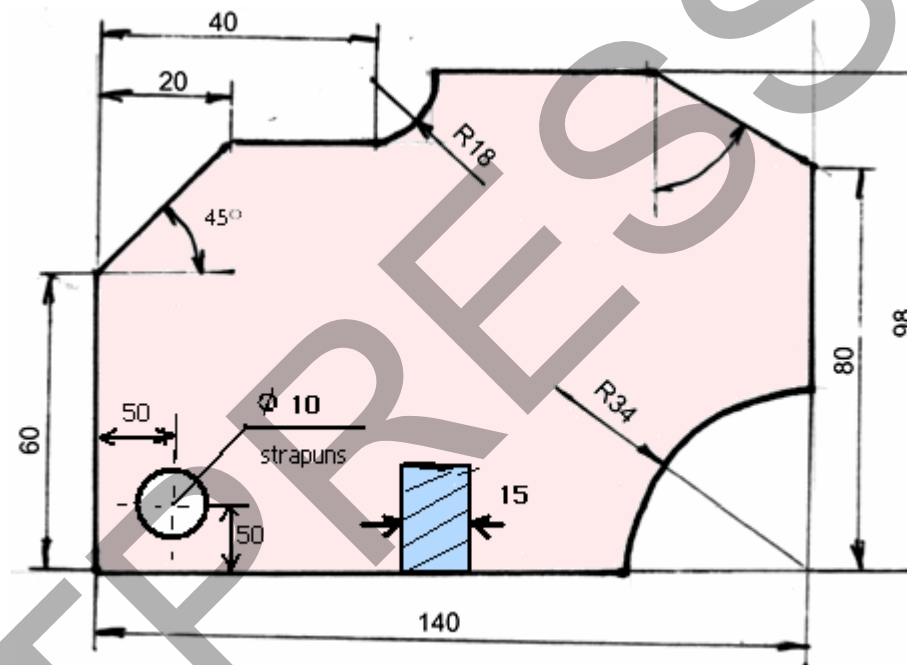


Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Se dă:



Realizarea piesei pe un sistem CNC presupune:

- Proiectarea unei tehnologii adecvate (faze, operații, scule, regim de așchiere)
- Efectuarea de către sculă a deplasărilor care se impun.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## Codificări referitoare la tehnologie:

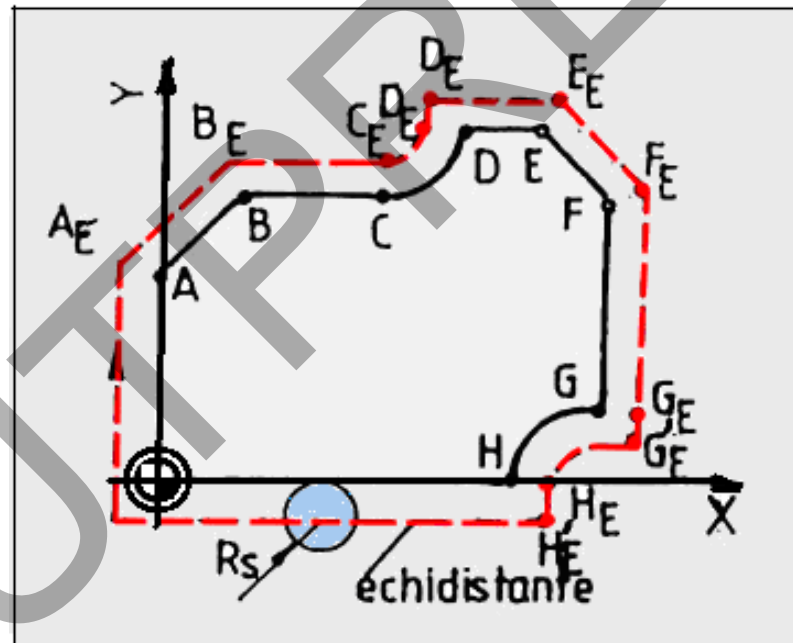
Scula: T urmat de un număr (identificarea sculei);

Turația AP: S urmat de un număr (valoarea/codul turației);

Avasul: F urmat de un număr (valoarea avansului)

Informații comandă pentru deplasare: X,Y,Z,A,B,C, etc.

Traectoria sculei: se programează echidistantă/conturul piesei



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

**Obs.: Fiecare porțiune de curbă se descrie într-un bloc separat**

Ca urmare programul sursă de prelucrare va conține o succesiune de blocuri pentru deplasarea sculei și pentru implementarea tehnologiei.

Exemplu:

(%)

0 0001 (număr program)

N0005 T01 M06 LF (schimbare sculă, T01 - freză);

N0010 G54 G90 S400 M03 LF (selectare sistem de coordonate, programare absolută, pornire AP cu 400 de rot/min)

N0015 G00 X-10.0 Y-10.0 LF (deplasare în punctul de start, Rf=10mm)

N0017 G43 Z-5.0 H01 M08 LF (activare corecție de lungime sculă, poziționare la z = - 5 mm, pornire lichid de răcire)

N0020 G01 Y64.142 F100 LF (deplasare cu avans de lucru w = 100 mm/min)

:

N80 G01 X-10.0 LF

N83 G91 G28 Z0 M19 LF (revenire la locul de schimbare a sculei oprire orientată AP)

N84 M01 (stop opțional)

N85 T02 M06 (schimbare sculă T02, burghiu de centruire)

N90 G54 G90 S600 M03 LF

:

N105 G91 G28 Z0 M19

M110 M30 LF (sfârșit program).



# Funcții G pentru operații de frezare

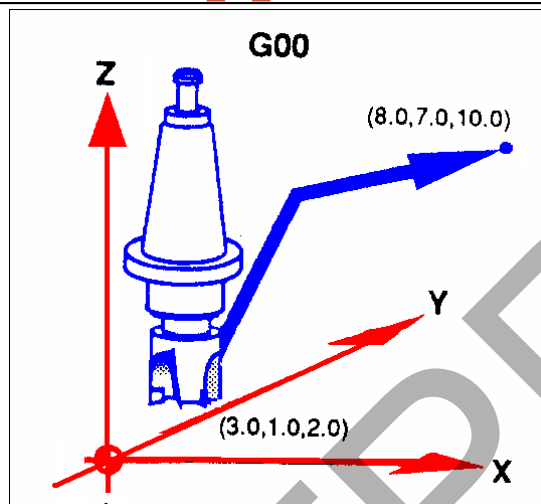
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## G00 Poziționare

G00 X\_ Y\_ :



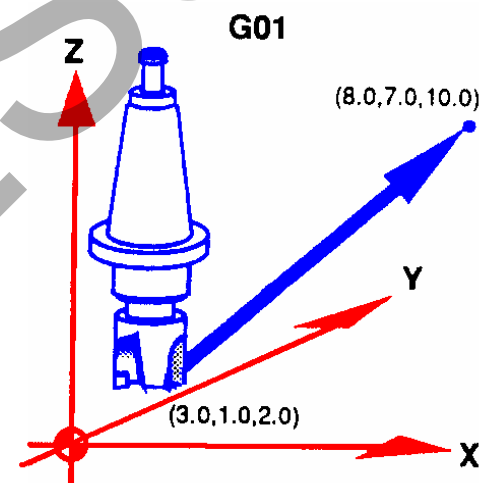
Formatul liniei de program (exemplu)

N60 G90 G00 X8.0 Y7.0 Z10.0

N60 denumire bloc  
G90 cotare absolută  
X coordonatele  
Y punctului  
Z final

## G01 Interpolare liniară (avans de lucru)

G01 X\_ Y\_ F\_ :



Formatul liniei de program (exemplu)

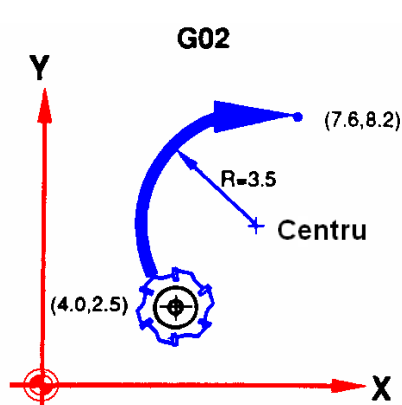
N60 G90 G01 X8.0 Y7.0 Z10.0

N70 denumire bloc  
G9 cotare incrementală  
G0 interpolare liniară  
X coordonatele  
Y punctului  
Z final  
F avans

## G02 Interpolare circulară (sensul acelor de ceasornic)

G02 X \_ Y \_ R \_ F \_ : Raza și poziția finală

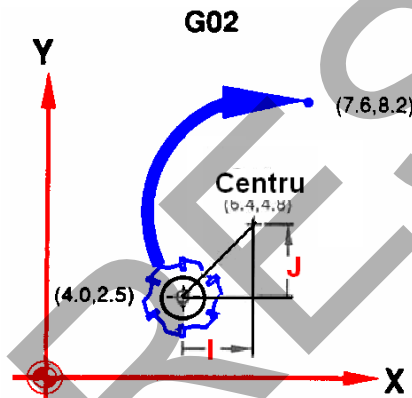
G02 X \_ Y \_ I \_ J \_ F \_ : Parametrii de interpolare și poziția finală



Formatul liniei de program  
(raza și poziția finală exemplu)

**N80 G90 G02 X7.6 Y8.2 R3.5 F100**

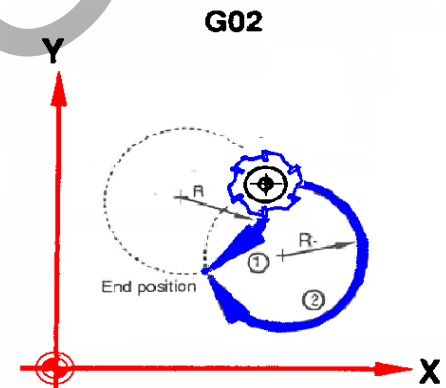
N80 denumire bloc  
G90 cotare absolută  
G02 interpolare circ. CLW  
X coordonatele punctului  
Y final  
R raza cercului  
F avans



Formatul liniei de program  
(param. de interpolare și poz. finală ex.)

**N90 G90 G02 X7.6 Y8.2 I2.4 J2.3F100**

N90 denumire bloc  
G90 cotare absolută  
G02 interpolare circ. CLW  
X coordonatele punctului  
Y final  
I param.interpolare  
J  
F avans



Formatul liniei de program  
(raza și poziția finală ex.)

**N100 G02 X \_ Y \_ R \_ F \_**

Raza și poziția finală cu arcul  
mai mic de 180°

**N110 G02 X \_ Y \_ R - F \_**

Raza și poziția finală cu arcul  
mai mare de 180°

Cap.1.  
Introducere

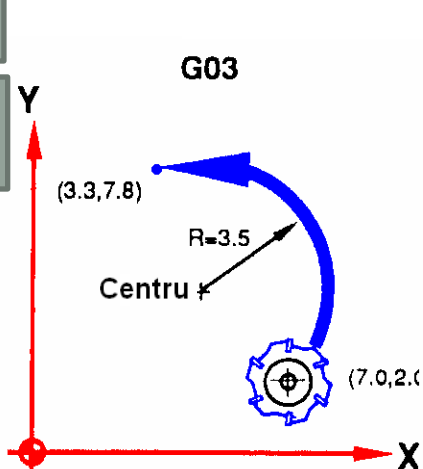
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

## G03 Interpolare circulară (sens contrar acelor de ceasornic)

G03 X \_ Y \_ R \_ F \_ : Raza și poziția finală

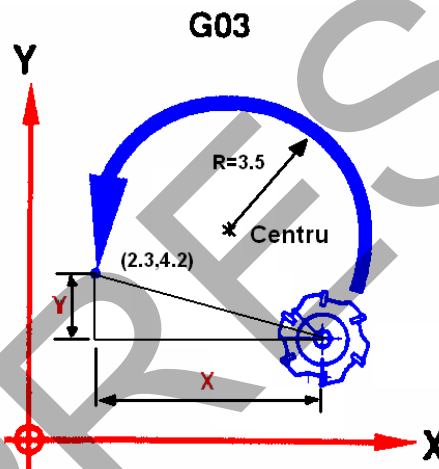
G03 X \_ Y \_ I \_ J \_ F \_ : Parametrii de interpolare și poziția finală



Formatul liniei de program  
(raza și poziția finală exemplu)

N120 G90 G03 X3.3 R3.5 F100

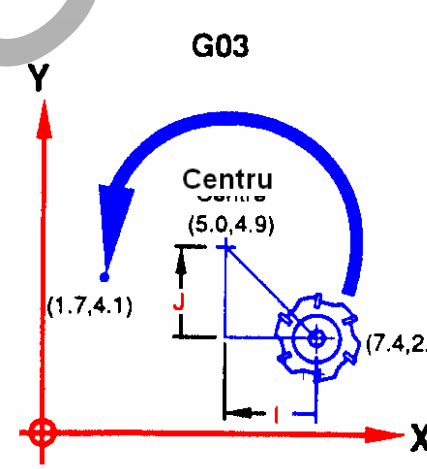
N120 denumire bloc  
G90 cotare absolută  
G03 interpolare circulară  
X coordonatele punctului  
Y final  
R raza cercului



Formatul liniei de program  
(raza și poziția finală exemplu)

N130 G91 G03 X5.7 Y1.5 R3.5 F100

N130 denumire bloc  
G91 cotare incrementală  
G03 interpolare circulară  
X coordonatele punctului  
Y final  
R raza cercului



Formatul liniei de program  
(param. de interpolare și poz.  
finală ex.)

N140 G90 G03 X1.8 Y4.I-2.4 J1.5 F100

N140 denumire bloc  
G90 cotare absolută  
G03 interpolare circulară  
X coordonatele punctului  
Y final  
I,J param.interpolare





## G04 Temporizare

Cap.1.  
Introducere

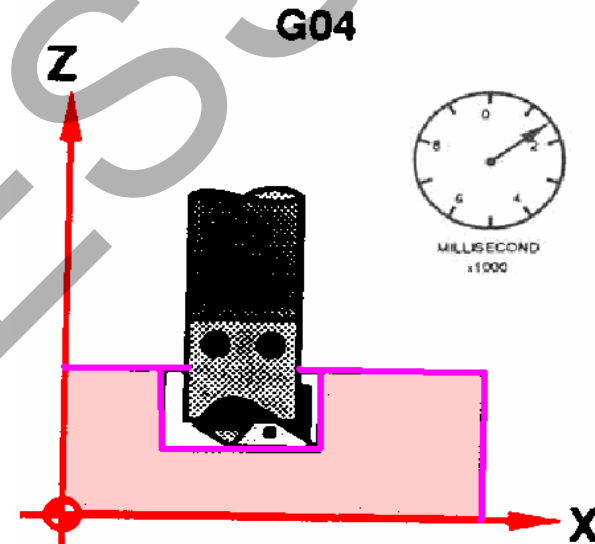
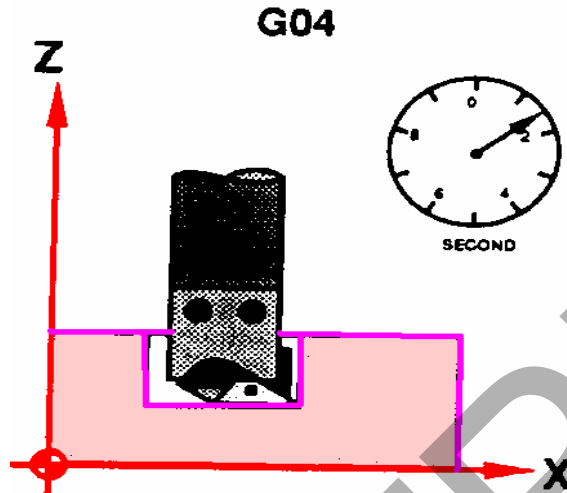
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

G04 X\_ : (secunde)

G04 P\_ : (milisecunde)

Observații: Pot fi și alte adrese (F)



Formatul liniei de program  
(alt exemplu)

```
N150 G04 X1.5;
```

N150 denumire bloc  
G04 temporizare  
X timpul de întrerupere (1,5 s)

```
N160 G04 P1500;
```

N160 denumire bloc  
G04 temporizare  
P1500 timpul de întrerupere (1,5 s)



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

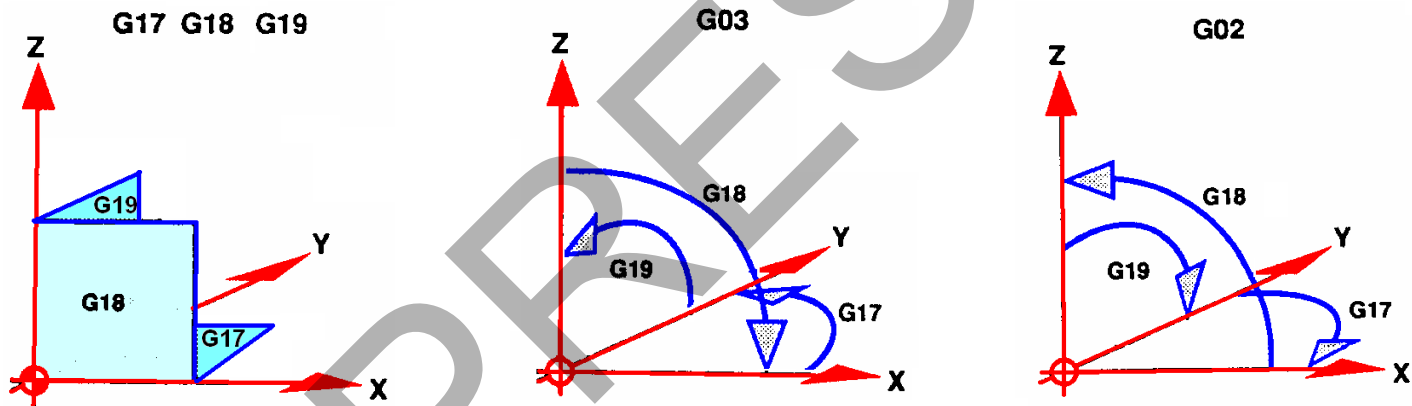
## G17

## G18

## G19

(Selectare plan de interpolare circulară și CR activă)

G17 : X – Y      planul XOY  
G18 : X – Z      planul XOZ  
G19 : Y – Z      planul YOZ



Formatul liniei de program

N170 G17 G02 X\_Y\_R\_F\_

N180 G17 G02 X\_Y\_I\_J\_F\_

N190 G18 G02 X\_Z\_R\_F\_

N200 G18 G02 X\_Z\_I\_K\_F\_

N210 G19 G02 Y\_Z\_R\_F\_

N210 G19 G02 Y\_Z\_I\_K\_F\_

Formatul liniei de program

N230 G17 G03 X\_Y\_R\_F\_

N240 G17 G03 X\_Y\_I\_J\_F\_

N250 G18 G03 X\_Z\_R\_F\_

N260 G18 G03 X\_Z\_I\_K\_F\_

N270 G19 G03 Y\_Z\_R\_F\_

N280 G19 G03 Y\_Z\_I\_K\_F\_



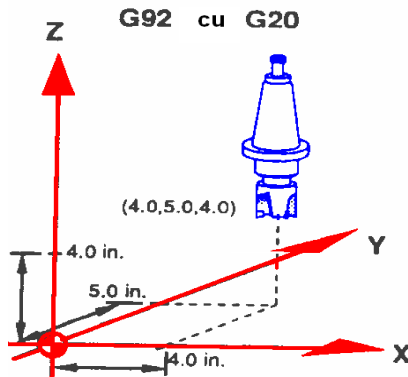
# G20, G21 Selectare sistem de măsurare

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

G20 :      inch  
G21:      mm

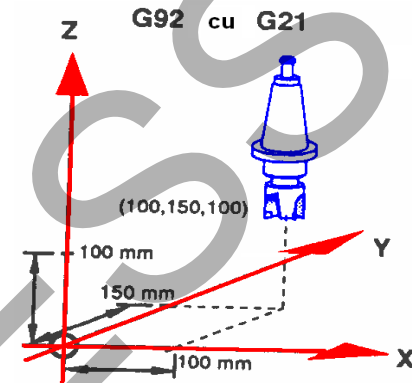
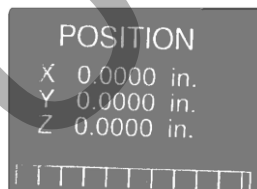


Formatul liniei de program  
(exemplu în inci)

```
N30 G20
```

```
N40 G92 X4.0 Y5.0 Z4.0
```

N30 denumire bloc  
G20 selectare „inch”  
N40 .....  
G92 deplasarea originii piesei



Formatul liniei de program  
(exemplu în milimetri)

```
N30 G21
```

```
N40 G92 X100 Y150 Z100
```

N30 denumire bloc  
G21 selectare „mm”  
N40 .....  
G92



## G22, G23 Memorare limite piese

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

G22 X \_ Y \_ Z \_ I \_ J \_ K \_

Prin parametrul de mașină se definește interiorul / exteriorul  
(RWL = 0 sau RWL = 1)

G23 – anulare G22

Formatul liniei de program (RWL parametrul 0 exemplu)

N470 G22 X100 Y70 Z0 I250 J210 K 100

N470 denumire bloc

G22 memorare limite piesă

X coordonatele punctelor

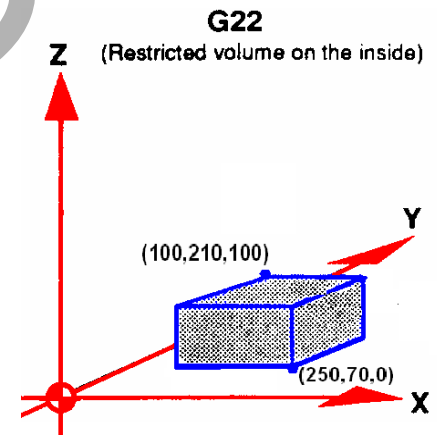
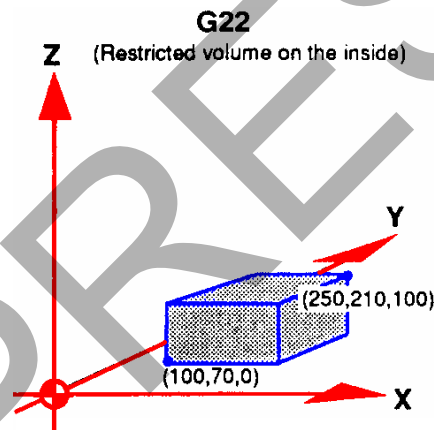
Y extreme

Z

I coordonatele punctelor

J minime

K



Formatul liniei de program (Anulare)

N490 G23 G00 X50 ;

N490

G23 anulare memorare limite

G00

X50

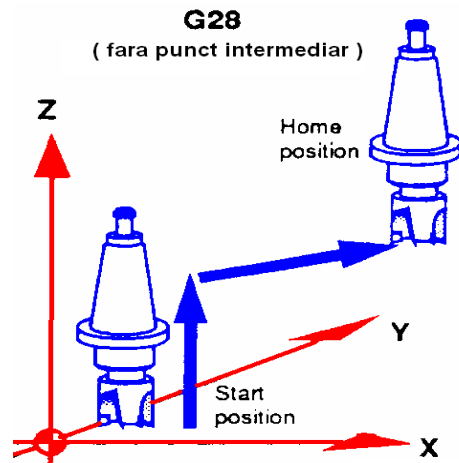


Cap.1.  
Introducere

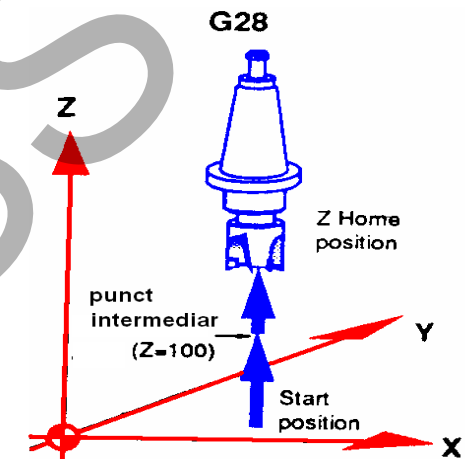
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## G28 Deplasare în punctul de referință (Home position)



punct intermediar



Formatul liniei de program  
(începutul programului exemplu)

```
N10 G91 G28 Z0.0
```

```
N20 G91 G28 X0.0 Y0.0
```

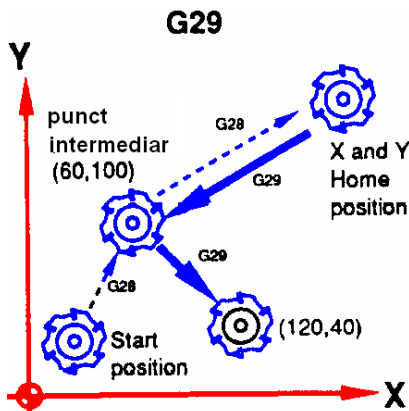
Formatul liniei de program

```
N400 G90 G28 Z100
```



## G29 Revenire din punctul de referință

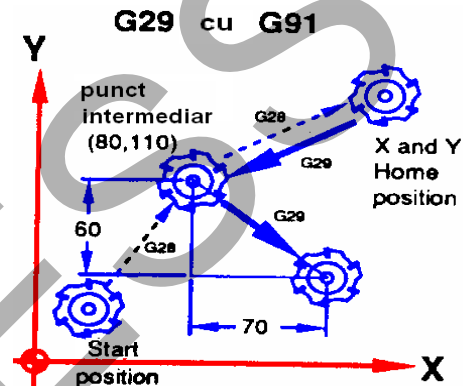
**G29 X \_ Y \_ Z \_ :**



Formatul liniei de program  
(coordonate absolute exemplu)

```
N410 G90 G28 Y100 ;
```

```
N420 G29 X120 Y40 ;
```



Formatul liniei de program  
(coordonate relative / incrementale)

```
N430 G90 G28 X80 Y110 ;
```

```
N440 G91 G29 X70 Y-60 ;
```

### **Comentariu:**

Scula este deplasată în punctul de referință via un punct definit prin X, Y, Z (G90 sau G91). Se programează la început de program pentru deplasarea sculei în PR sau schimbarea sculei.

Se recomandă ca prima deplasare să fie după Z urmată de deplasarea după X, Y.

Măsură de siguranță.



Cap.1.  
Introducere

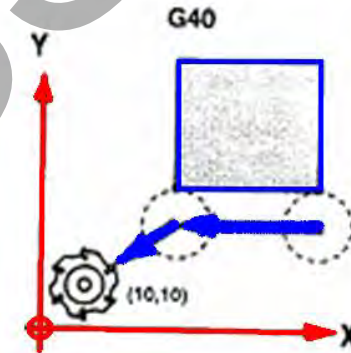
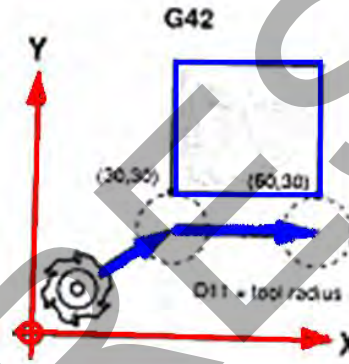
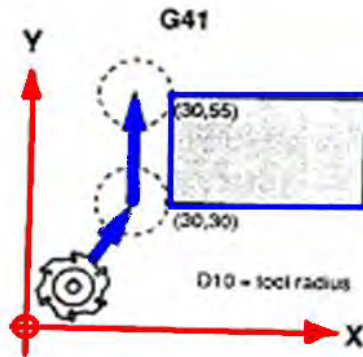
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

## G40, G41, G42. Corecția de rază

G40      Anulare  
G41      CR,  
G42      CR,

scula pe stânga  
scula pe dreapta



## Formatul liniei de program

N270 G90G17 G41 D10 X30 Y30 F100;

N290 G90 G17 G01 G42 D11 X30 Y30 F100

N310 G00 G40 X10 Y10

N280 G01 Y55 ;

N300 G01 X60

## G43, G44, G49 Corecția de lungime, CL

G43 : CL „+”

N920 G49 ;      Anulare CL

G44 : CL „-”

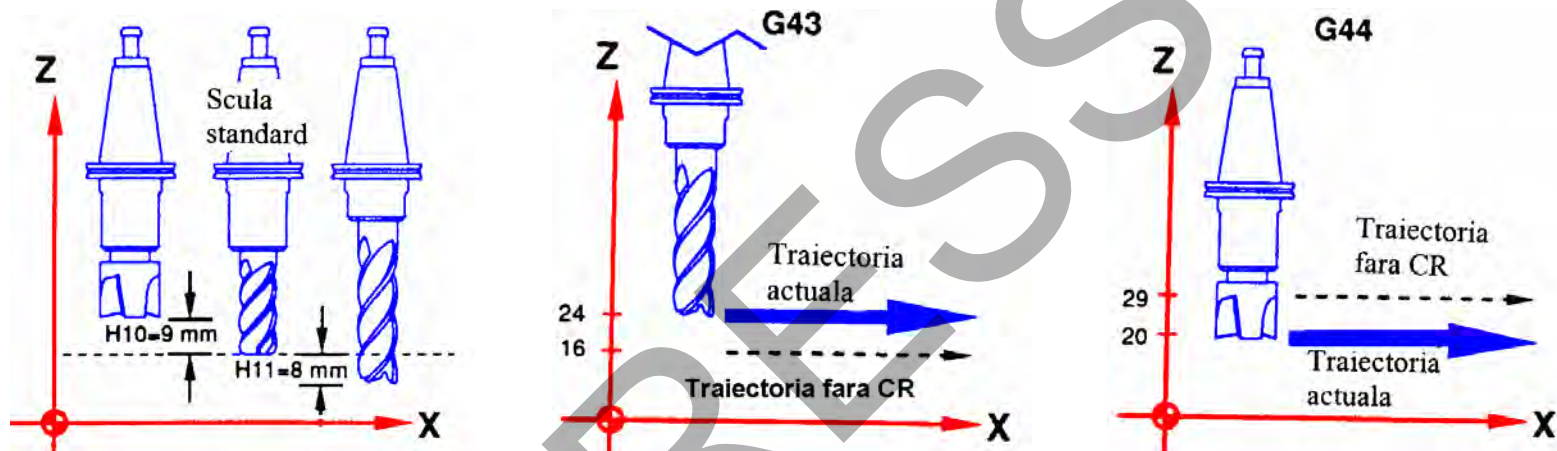
N920 H00 ;      Anulare CL



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO



## Formatul liniei de program

```
N900 G43 Z_ H11;
```

```
N910 G44 Z_ H10;
```

S-a prezentat varianta cu registrul „H”. La majoritatea ECN, valoarea corecției de lungime se memorează în registrul D... CL la CNC.



## Declarare sistem de coordonate.

### G54, G55, G56, G57, G58, G59

G54 – prima deplasare de origine

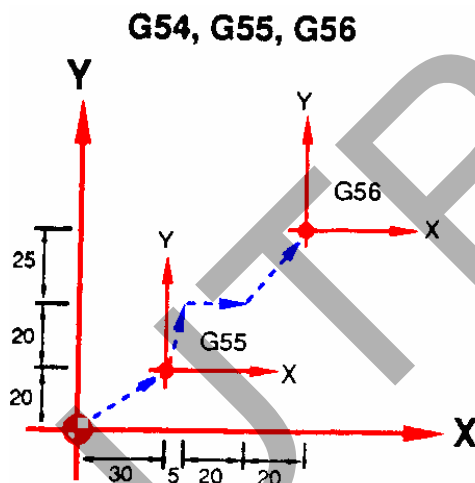
.

.

G59 – a șasea deplasare de origine

Regim: OFFSET pe panoul echipamentului

Deplasările sunt realizate în varianta incrementală (G91)



Formatul liniei de program

```
N460 G55 G00 X30 Y20 ;
```

```
N460 G55 G00 X30 Y20 ;
```

```
N460 G55 G00 X30 Y20 ;
```

```
N460 G55 G00 X30 Y20 ;
```

Valorile X, Y, Z sunt date considerând poziția curentă a sculei



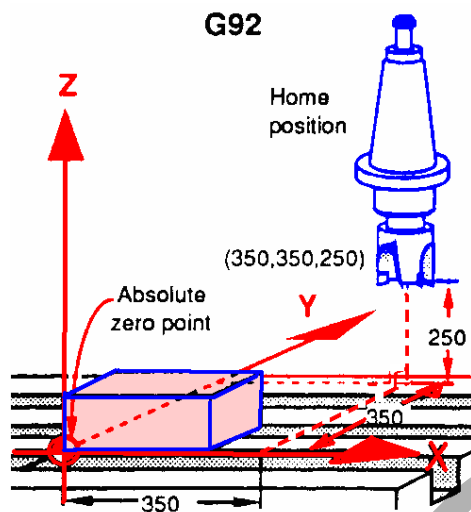
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

## G92 Programare suplimentară de origine

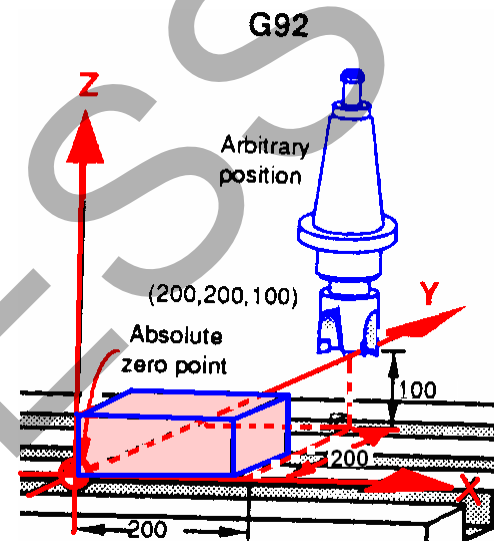
G92 X \_ Y \_ Z \_ :



Formatul liniei de program  
(poziția de 0 exemplu)

```
N40 G91 G28 X0 Y0 Z0 ;
```

```
N80 G92 X350 Y350 Z250
```



Formatul liniei de program  
(poziție arbitrară exemplu)

```
N100 G92 X200 Y200 Z100 ;
```

**Comentariu:** Adresele X, Y, Z, definesc poziția curentă a sculei în raport cu noua origine.

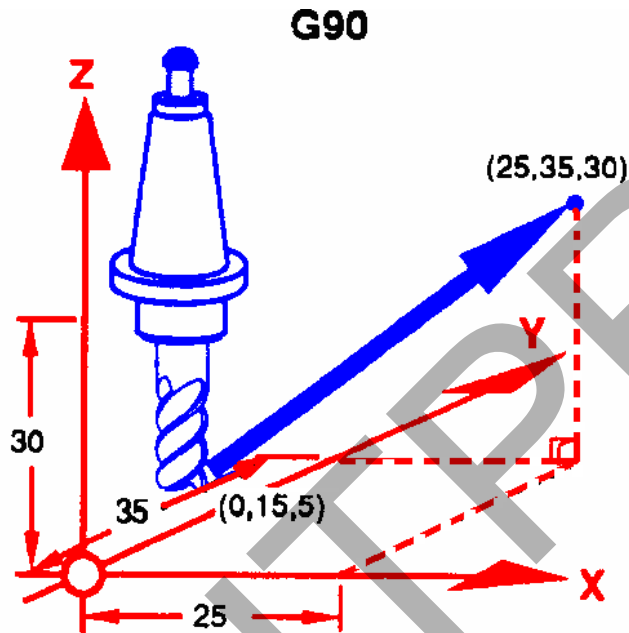


Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

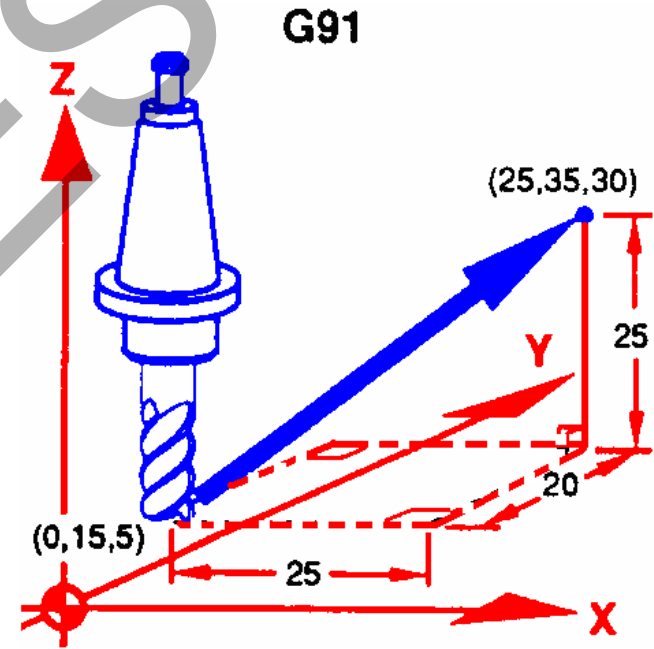
Cap.3.  
Limbaj ISO

## G90 Sistem absolut de programare



N50 G90 G01 X25 Y35 Z30

## G91 Sistem incremental de programare



N60 G91 G01 X25 Y20 Z25 F100



# Cap.4. PROGRAMAREA DEPLASĂRILOR

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

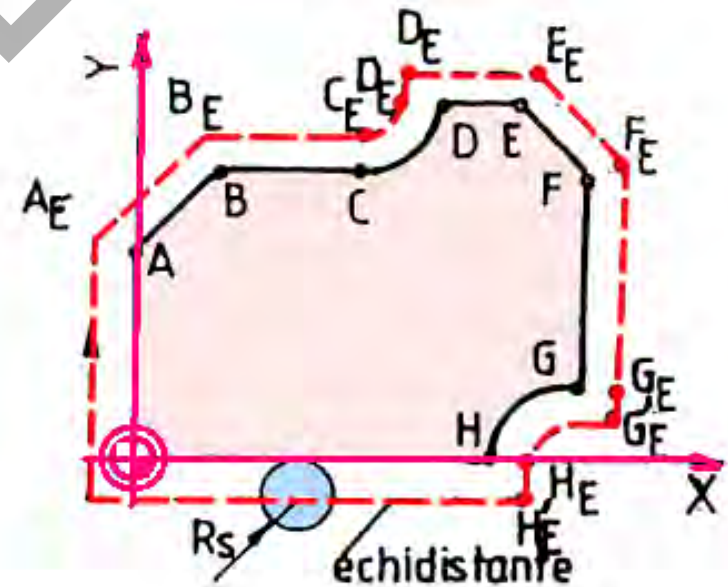
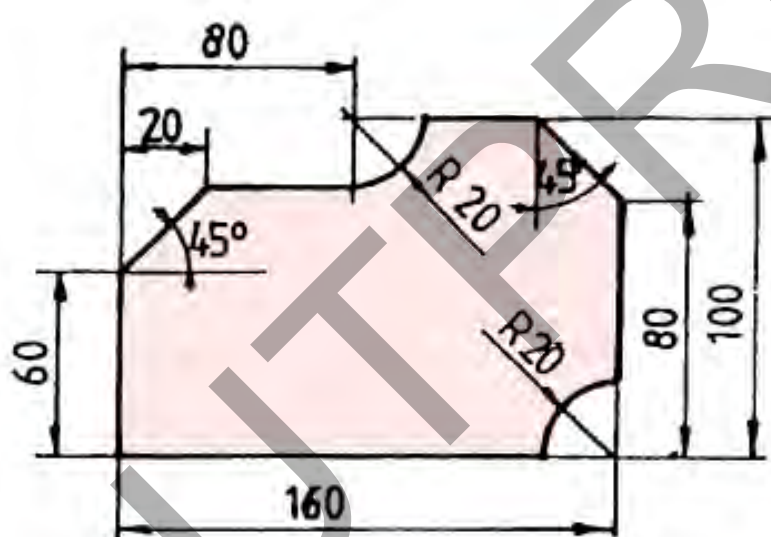
## Aspecte generale

Programarea traiectoriei sculei, în vederea realizării unei piese, se poate face în 2 feluri:

- Se programează direct conturul piesei. Corecția de rază
- Se programează echidistantă

Echidistantă

A, B, C, Puncte de contur



Cap.1.  
Introducere

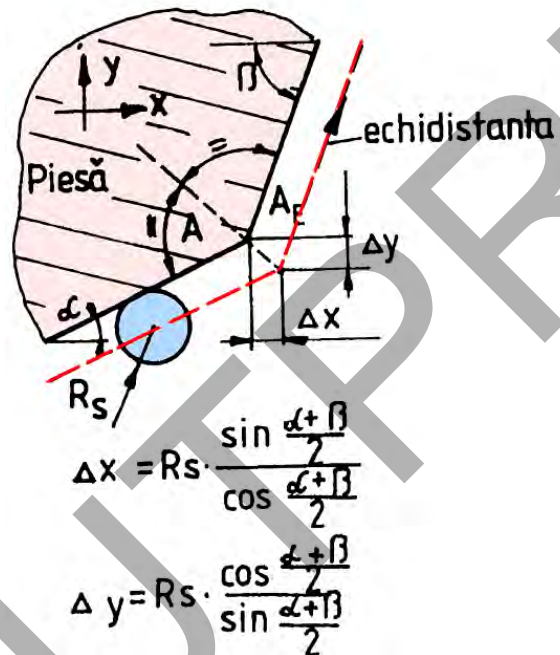
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

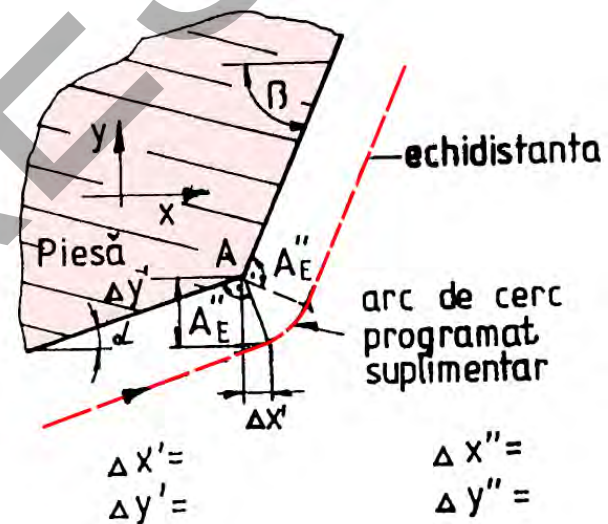
Cap.4.  
Programarea  
deplasarilor

## Calcul:

### Metoda Bisectoarei



### Metoda arcului de cerc



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

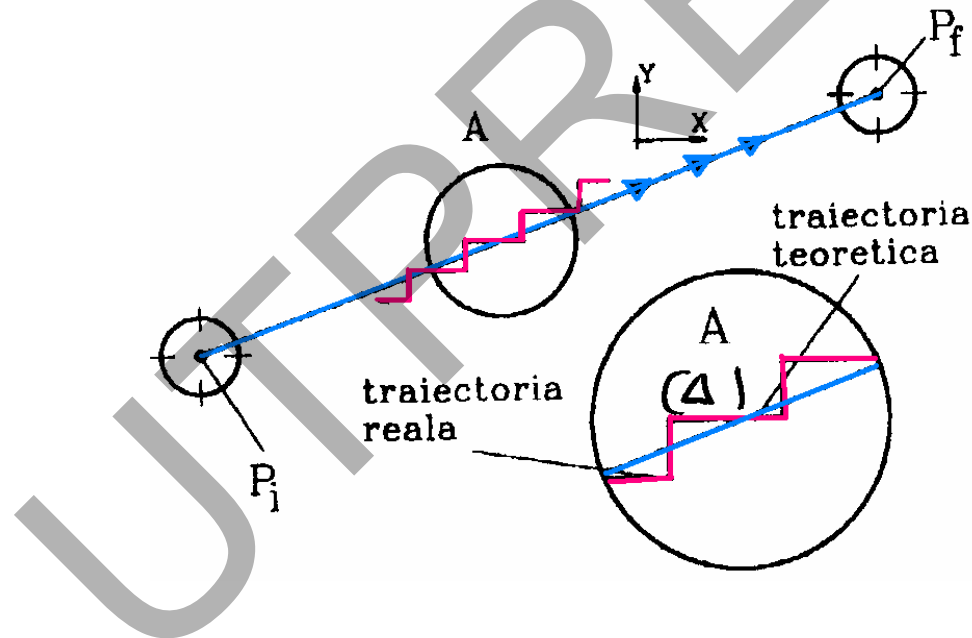
Cap.4.  
Programarea  
deplasarilor

# Interpolare

Conform DEX ed. 1984; a interpola: „a intercala într-un șir de valori cunoscute una sau mai multe mărimi determinate sau estimate”.

Mărimea pașilor ( $\Delta$ ) : - BLU

- prec. s. m.



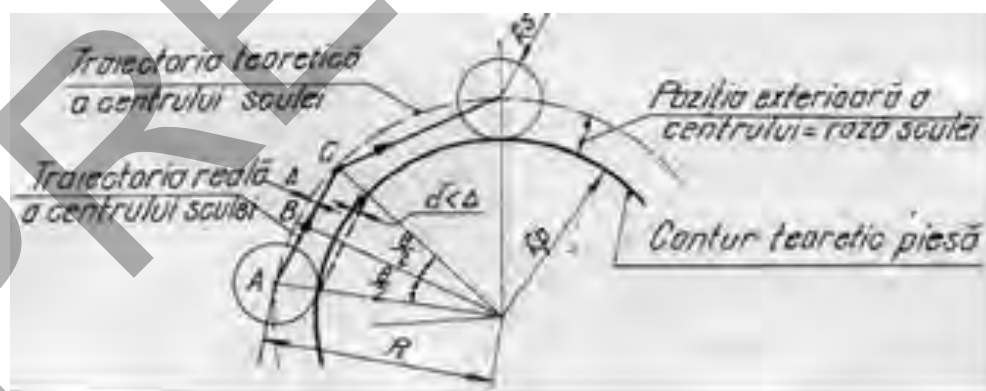
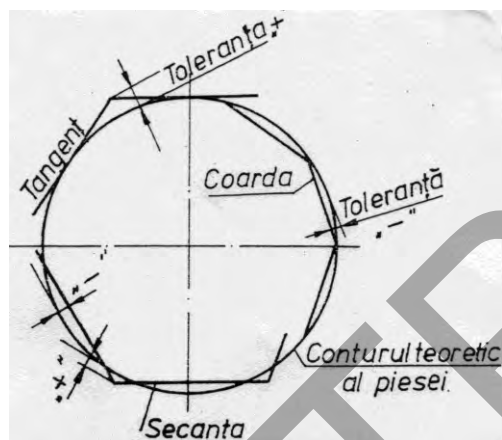
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

## Aproximarea curbelor



$$AC=2 \times AB \quad AB=\sqrt{2R\Delta}$$

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

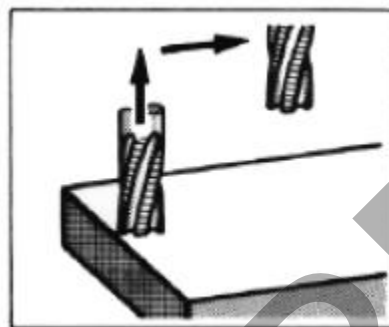
Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

# Controlul deplasării sculei

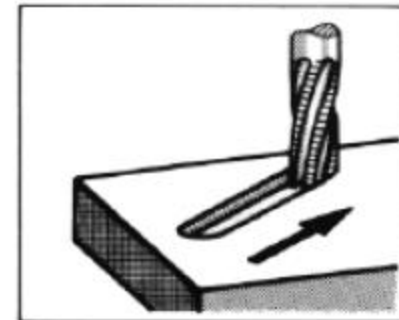
G00

Deplasare cu  
avans rapid



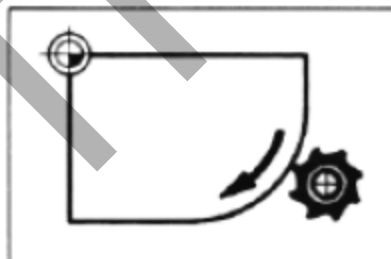
G01

Interpolare  
liniară



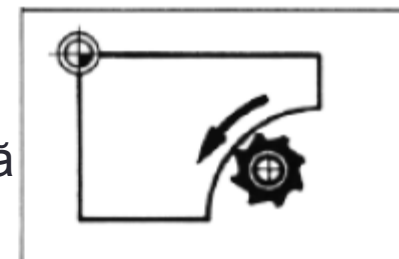
G02

Interpolare  
circulară  
CLOCKWISE



G03

Interpolare circulară  
COUNTERCLOCK  
WISE





Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Deplasare cu avans rapid G0

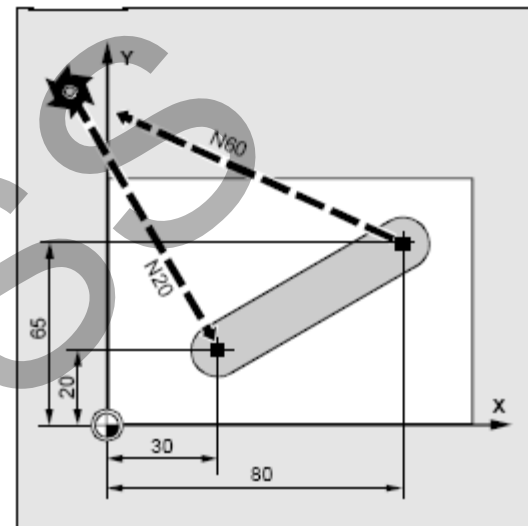
Programare

G0 X...Y...Z...

G0 AR=...RP=...

Semnificația parametrilor

XYX	Coordonate carteziene punct final
AP=	Coordonate polare unghi
RP=	Coordonate polare rază



N10 G90 S400 M3	Sistem absolut pornire AP sens orar
N20 G00 X30 Y20 Z2	Poziționare în punctul de start
N30 G01 Z-5 F1000	
N40 X80 Y65	
N50 G00 Z2	Retragere la Z=2 mm
N60 G00 X-20 Y100 Z100 M30	Retragere în punctual inițial



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

## Interpolare liniară G1

Programare

G1 X... Y... Z... F...

G1 AP=... RP=... F...

Semnificația parametrilor

Semnificația parametrilor	
X Y Z	Coordonate carteziene punct final
AP=	Coordonate polare unghi
RP=	Coordonate polare rază
F	Viteză de avans
N10 G90 S400 M3	
N20 G0 X30 Y20 Z2	
N30 G01 Z-5 F1000	Deplasare cu avans de lucru la z=5
N40 X80 Y65	Deplasare în punctul final
N50 G0 Z2	
N60 G0 X-20 Y100 Z100	



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

## Interpolare circulară G2/G3, CIP

Programare

G2/G3 X...Y...Z...I...J...K...

G2/G3 AP=... RP=...

G2/G3 X...Y...Z...CR=...

G2/G3 AR=...I...J...K...

G2/G3 AR=...X...Y...Z...

CIP X...Y...Z...I1=...J1=...K1=...

CT X...Y...Z...

### Semnificația parametrilor și a comenzilor

<b>G2</b>	Deplasare circular în sens orar
<b>G3</b>	Deplasare circular în sens antiorar
<b>CIP</b>	Deplasare circulară printr-un punct intermediar
<b>CT</b>	Cerc cu tranziție tangențială
<b>X Y Z</b>	Coordonatele punctului final în sistem cartezian
<b>I J K</b>	Coordonatele centrului cercului în sistem cartezian
<b>AP=</b>	Coordonatele punctului final în sistem polar, în acest caz unghiul polar
<b>RP=</b>	Coordonatele punctului final în sistem polar, în acest caz raza cercului
<b>CR=</b>	Raza cercului
<b>AR=</b>	Unghiul arcului de cerc
<b>I1= J1= K1=</b>	Coordonatele punctului intermediar în sistem cartezian

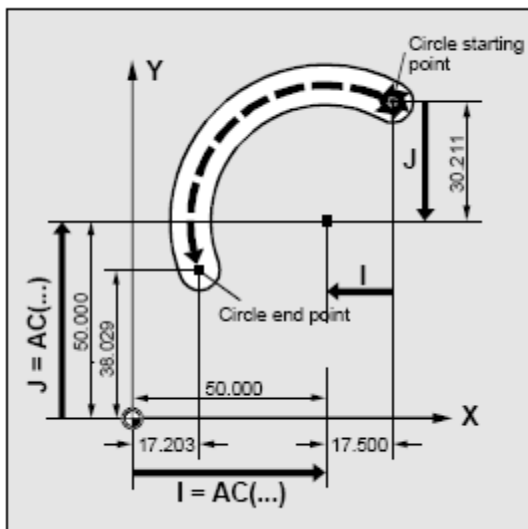


Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor



I,J,K Programare în sistem incremental

```
N10 G0 X67.5 Y80.211
```

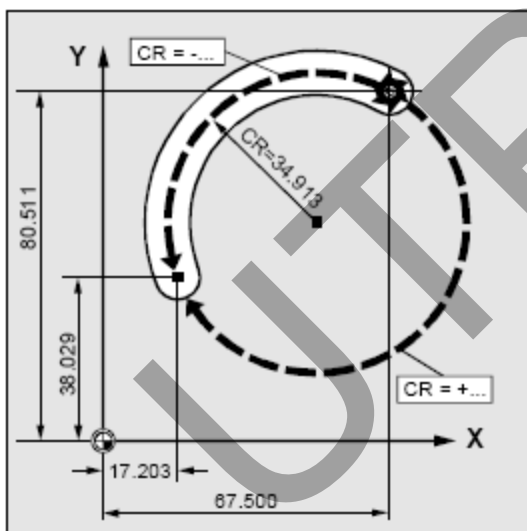
```
N20 G03 X17.203 Y38.029 I-17.5 J-30.211 F500
```

I,J,K Programare în sistem absolut

I=AC(..) J=AC(...) K=AC(..)

```
N10 G00 X67.5 Y80.211
```

```
N20 G03 X17.203 Y38.029 I=AC(50) J=AC(50)
```



```
N10 G00 X67.5 Y80.211
```

```
N20 G03 X17.203 Y38.029 CR=34.913 F500
```

CR=- pentru unghi la centru  $\leq 180^\circ$

CR=+ pentru unghi la centru  $> 180^\circ$



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasărilor

## - Utilizarea „R”

N30 G00 X20.0 Y50.0 LF - poz. în punctul A

N35 G02 X100.0 Y50.0 R50.0 F100 LF interpolare circulară

Observații.: În blocul N35 Y poate lipsi fiind același cu Y din blocul N30.

**NU** toate echipamentele au o astfel de facilitate!

## - Utilizarea I, J, K

Metoda cea mai veche

Pot fi: - scalari (ECN cu restricții)

- vectori (ECN fără restricții)

I – proiecția distanței : “ punct început - centru arc” pe axa X

J - proiecția distanței :” punct început - centru arc” pe axa Y

K - proiecția distanței : ” punct început - centru arc” pe axa Z

Programare greșită a parametrilor I, J, K produce:

- Eroare detectată de ECN prin punctul final. => ECN emite un semnal de avertizare
- Eroare nedetectată (se încadrează în toleranțe)
- cu I, J, K prea mari se produce sub tăierea cercului
- cu I, J, K prea mici se produce o umflare a cercul (scula în afara traiectoriei normale)

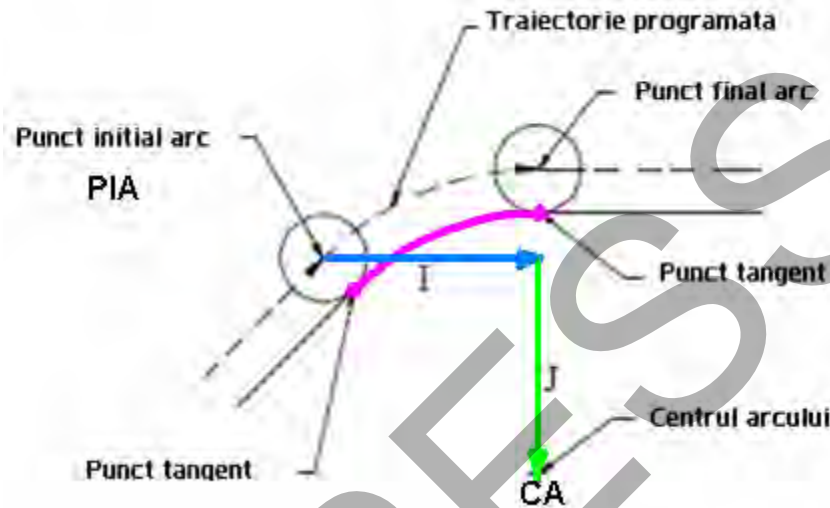


Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor



Există anumite dificultăți în a înțelege semnificația mărimilor I și J când de fapt sunt simple: Prin I și J se notează direcțiile de la punctul inițial al arcului de cerc la centrul arcului. Calcularea valorilor I și J

$$I = X_{centru.arc} - X_{punct.inceput.arc}$$

$$I = X_{CA} - X_{PIA}$$

$$J = Y_{centru.arc} - Y_{punct.inceput.arc}$$

$$J = Y_{CA} - Y_{PIA}$$

sau

I = distanța de la PIA la CA în direcția X

J = distanța de la PIA la CA în direcția Y

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

# Restricția de cadran

*Cadran?* Paralele cu axele sistemului de coordonate.

La ECN cu restricții în cadrul unui bloc se poate programa un arc de cerc care nu depășește limitele unui cadran trigonometric. I, J, K sunt distanțe.

## •Echipament cu restricții:

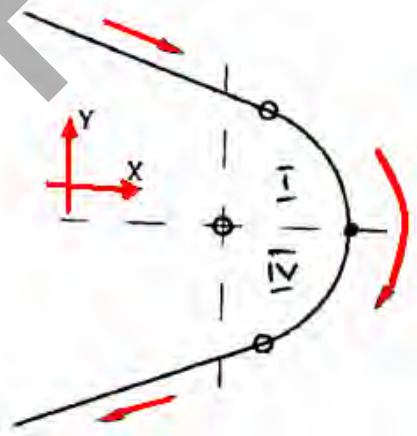
```

N20      G00      X20000  Y50000  LF
N25      G02      X60000  Y70000  I40000  J30000  F100  LF
N30      G02      X100000 Y50000  I0      J0      F100  LF
  
```

## •Echipamente fără restricții:

```

N20      G00      X20000  Y50000  LF
N25      G02      X100000 Y50000  I40000  J-30000  F100  LF
  
```



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

# Cap.5. INFORMAȚII TEHNOLOGICE

## Viteza de avans

$F$  \_\_\_\_\_ - Viteza de avans, mm/min

- Avansul, mm/rot sau  $\mu\text{m}/\text{rot}$

- Codul FRN  $\text{min}^{-1}$

-  $\text{FRN} = \text{viteza de avans}/\text{distanța} \times 10 [\text{min}^{-1}]$

- Codul magic Trei (MT)

(a- număr)  $a > 1$   $C_1 C_2 C_3 C_4 C_5 C_6$   $C_1 C_2 C_3$

$C_1 = 3 + n$

$C_2 C_3 = C_1 C_2$

$n$

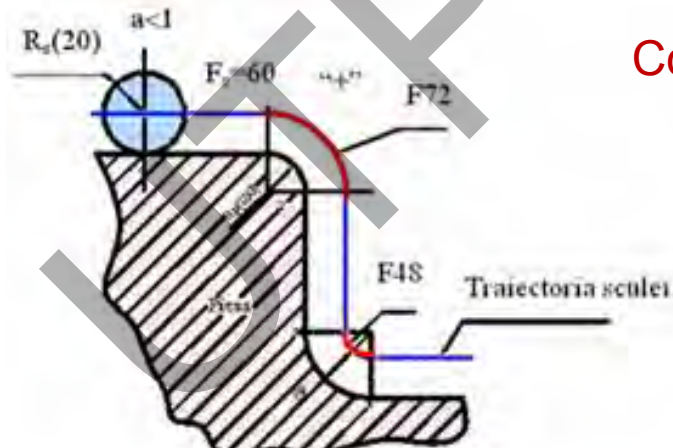
$a < 1$   $0,00 C_1 C_2 C_3$   $C_1 C_2 C_3$

$C_1 = 3 - n$

Corelarea vitezei de avans programată cu viteza reală de pe m-u

$C_2 C_3 = C_1 C_2$

Corelarea vitezei de avans programată  
cu viteza reală de pe m-u.



$$F = \frac{R_p \pm R_s}{R_p} F_c$$



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

# Selectare turație (S)

S \_\_ - codificare simbolică, **cod**

S \_ \_ \_ \_ - turația arborelui principal, **rot/min**

- viteza constantă de aşchiere, (specifică pentru strunguri) **m/min**

## Exemplu

S1250 : -1250 rot/min

S10 : a 10-a turație din gama de turații a mașinii

G96 S \_ \_ \_ \_ - menținerea constantă a vitezei de aşchiere

# Selectare sculă (T)

T \_\_

T \_ \_ \_ \_

## Exemplu:

T3 M6 pentru NC - schimbarea efectivă

CNC - schimbarea efectivă plus activare corecție de lungime

T8 - selectare sculă, fără schimbare efectivă



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informații  
tehnologice

# Programarea sculei

T\_\_ Cifrele indică numărul sculei / numărul poziției din magazinul de scule.

D\_\_ / H\_\_ offset scule (lungime și rază)

Notă!

- ❖ Pentru echipamente CNC (frezare) registrul D se utilizează pentru CR iar registrul H pentru CL;
- ❖ Pentru aplicații de strunjire, frecvent, scula se programează :

T\_\_ . \_\_

Primele două cifre specifică scula iar ultimele registrul de corecție.

Ex. T 09 16 în care:

09- număr sculă

16- număr registru de corecție

Comanda pentru schimbarea sculei, în mod normal, se programează în două etape:

- este selectată prima dată scula, din magazinul de scule;
- scula selectată este încărcată apoi în arborele principal al mașinii-unelte.

Cele două etape menționate sunt specifice numai centrelor de prelucrare.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Schimbarea sculei prin comanda T

Această posibilitate este implementată la strunguri cu magazine de scule de tip circular. La programarea adresei T se inițiază ciclul de schimbare a sculei, fără a fi nevoie și de programarea cuvântului NC "M06".

Sintaxa :

T <număr>            T10 număr: 0.....32000

T = <număr>        T = 10

T <n> = <număr>    T12 = 4

Anularea comenzi se face prin programarea: T0 sau T0 = <număr>

N10 T1 D1.....; încarcă scula # 1 și activează registrul D1

.

.

N60 T0 .....; anulează /deselectează scula T1;

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Schimbarea sculei utilizând M06

Selectarea sculei se face numai prin programarea adresei Tx iar schimbarea prin programarea Ty M06.

N10 T1 M6.....; încarcă scula T1;  
 N20 D1.....; selectează corecția de rază;  
 N30 G1 X200.....; prelucrare cu scula T1;  
 .....  
 N70 T5.....; preselectare scula T5;  
 N80 .....; prelucrare cu scula T1 (în  
 continuare);  
 .....  
 N100 M06.....; încarcă scula T5;  
 N110 G1X200 D5.....; prelucrare cu scula

# Schimbarea sculei cu managementul de scule

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

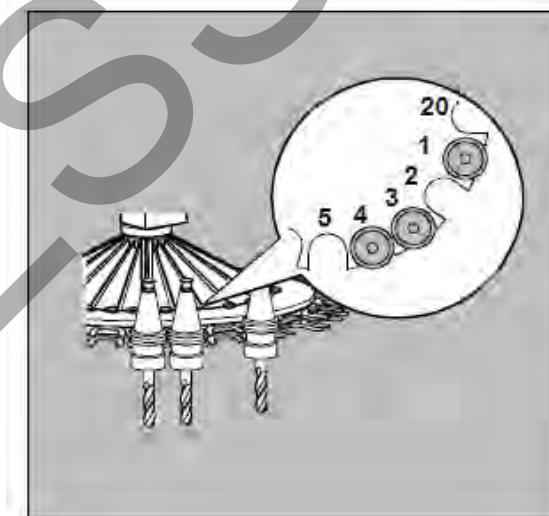
Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Apelarea sculei se face via denumirea sa: T="BURGHIU"

locați	scula	grup	stare
a			
1	Burghiu, duplicate, nr=1	T15	Blocată
2	Neocupată		
3	Burghiu, duplicate, nr=2	T10	Permis
4	Burghiu, duplicate, nr=3	T1	Activă
5..20	Neocupată		



Apelarea este procesată în felul următor:

1. se consideră locația 1 din magazin și se identifică scula;
2. managementul sculei recunoaște că scula este blocată și ca urmare nu poate fi utilizată;
3. se inițiază căutarea pentru T = "BURGHIU" în concordanță cu metoda de căutare, se găsește o sculă activă sau se selectează următoarea sculă duplicat.
4. sunt găsite următoarele scule folosibile: "BURGHIU" duplicate no.3 (în locația a 4-a a magaziei)
5. se inițiază schimbarea sculei

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Utilizarea corecțiilor de sculă (offseturi) Parametrii sculei

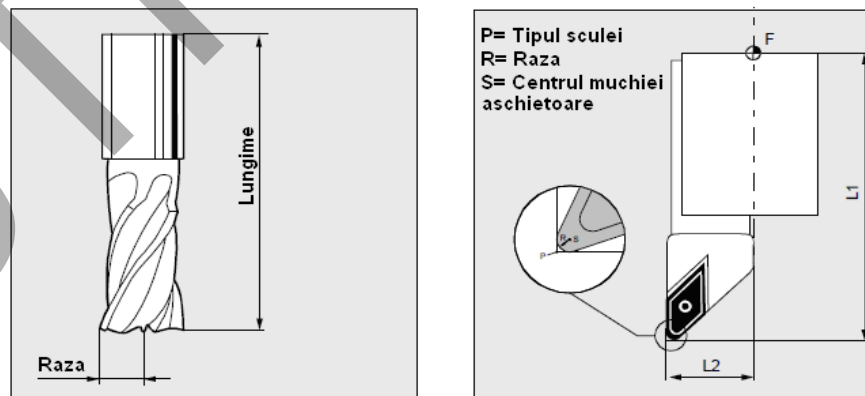
Geometria sculei din punct de vedere a programării este definită de:  
Lungime și diametru.

În cadrul programului se fac referiri la informațiile menționate prin  
programarea unor registre D/H.

Valoarea ce urmează a fi memorată în ele se introduce ulterior.

În cazul CNC-urilor, rolul regiștrilor de corecție este luat de fișierele cu  
sculele.

Alte informații memorate sunt cele referitoare la tipul sculei (burghiu,  
freză sau cuțit de strunjit) și direcția vârfului sculei pentru operații de  
strunjire.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

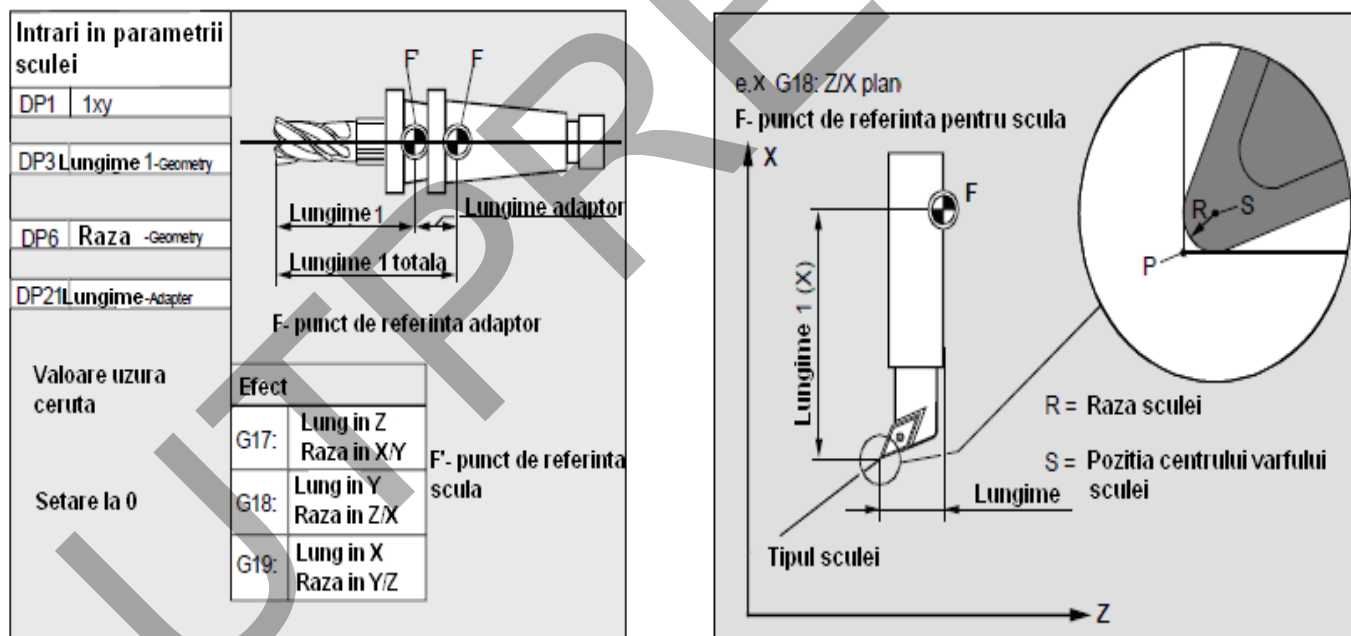
Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Pentru introducerea valorii parametrilor, echipamentele CNC au implementate o “listă a tipurilor de scule”. Parametrii relevanți ai sculei, în exemplul indicat, trebuie introduși în câmpurile respective prin “DP....”.

Parametrii care nu sunt necesari se setează “zero”



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

# Programarea corecțiilor de sculă

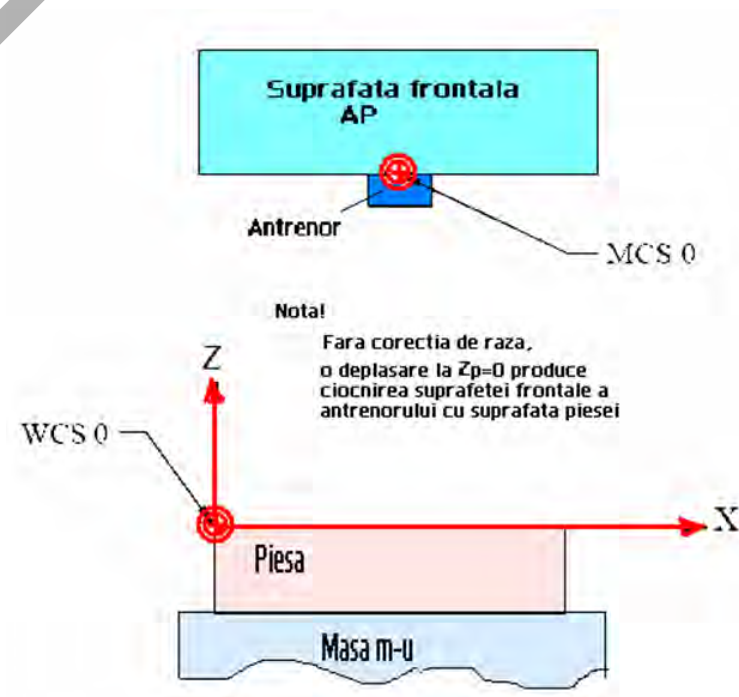
Corecția de lungime  
Corecția de masă (paraxială)  
Corecția de rază

Valoarea corecțiilor se programează în regiștrii D sau H (echipamente NC) sau în tabele de scule (echipamente CNC)

$$VD = V_p \pm VC$$

## Corelația de lungime a sculei (C.L.)

Rememorare: MCS este la capătul AP, traductorul pe axa Z măsoară distanța de la MCS la WCS astfel că mașina poate compensa deplasarea în raport de poziția piesei.





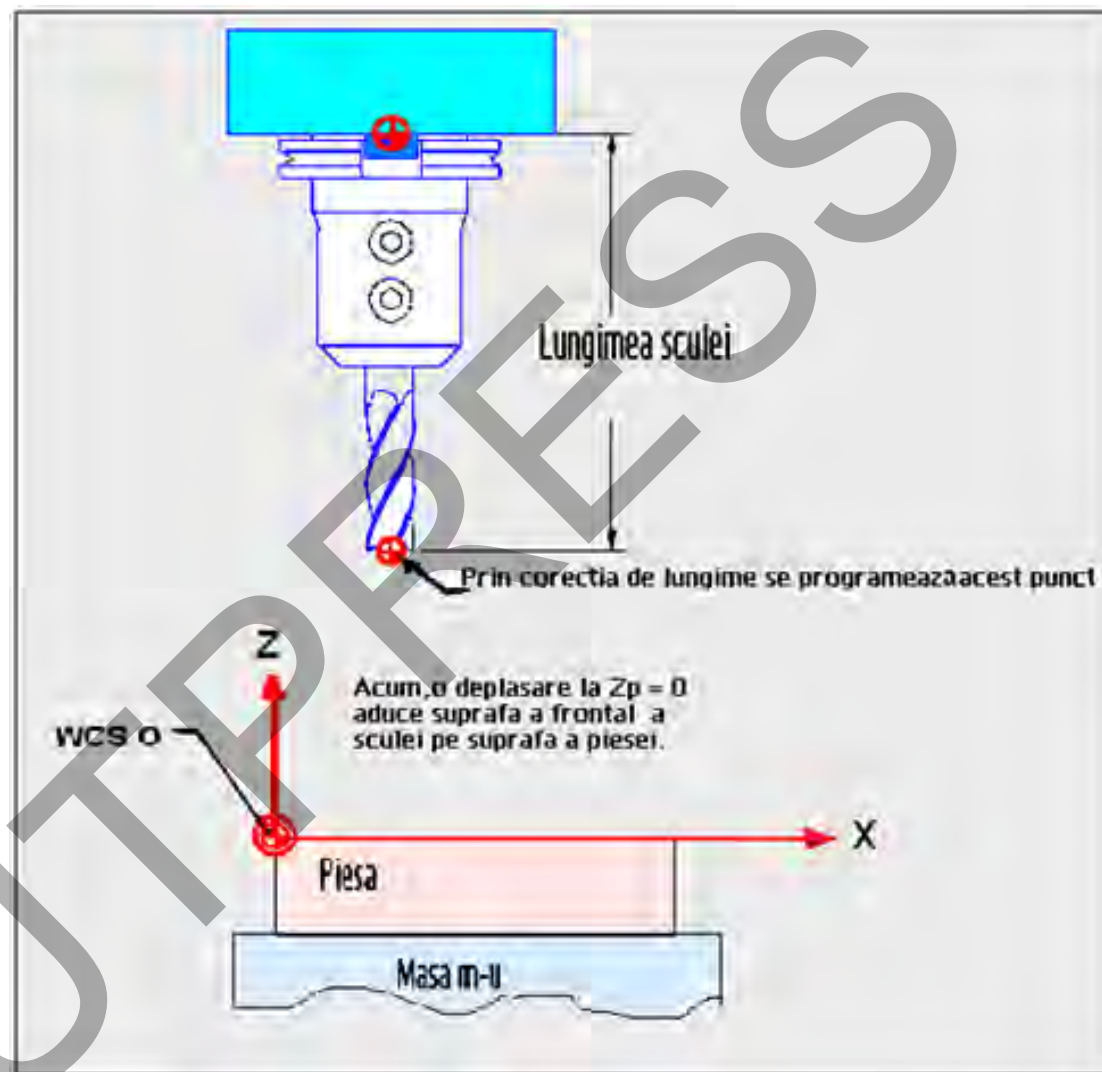
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice



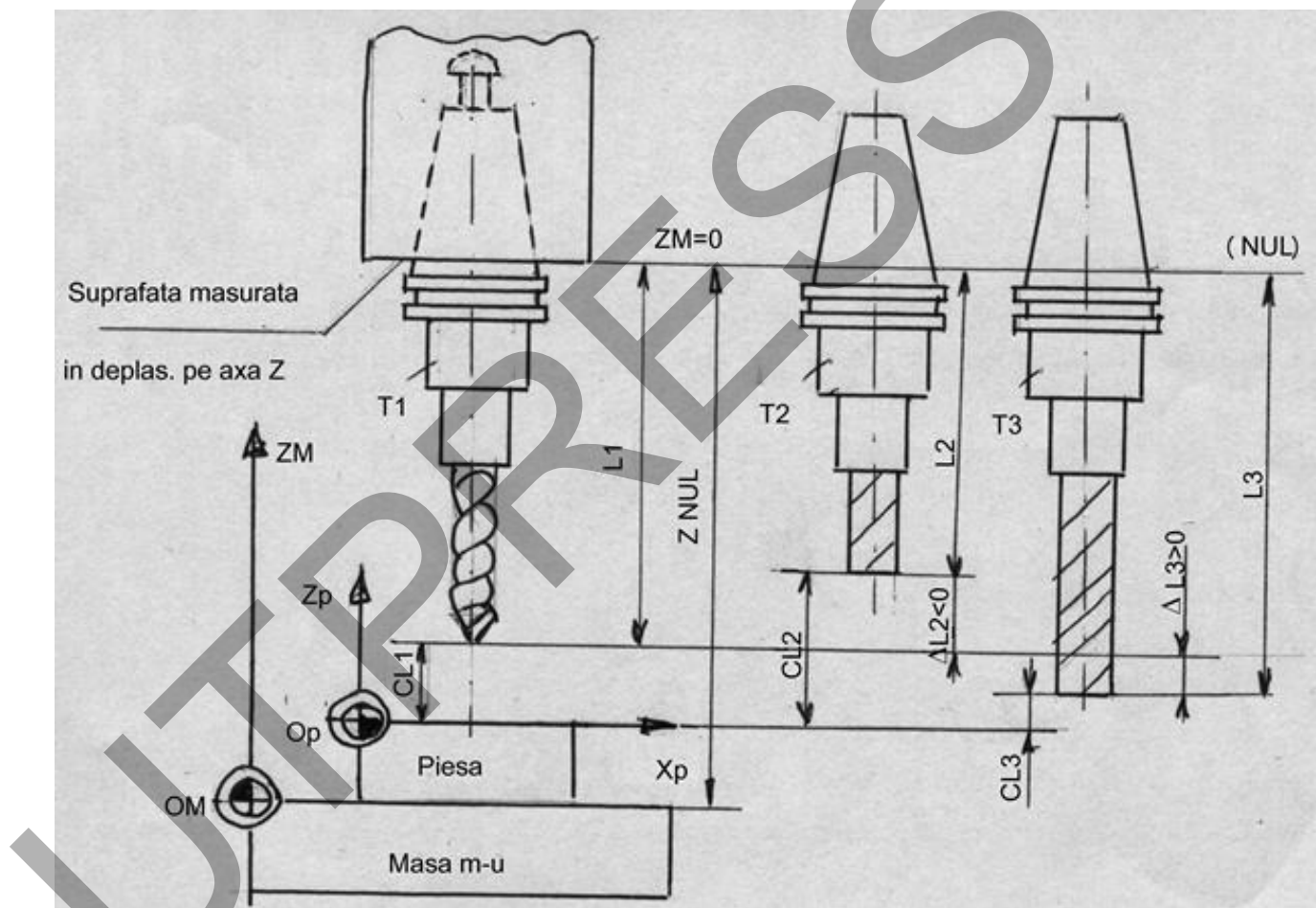
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Exemple de programare a corecției de lungime pe câteva echipamente numerice reprezentative sunt indicate în continuare:

- ❖ echipamente NC clasice, având adresele G43 și G44 implementate

N60	T1	S12	M3	M6	LF	Schimbare sculă
N65	G00	G44	Z15000	D1	LF	Corecția D1

activă

- ❖ echipamente NC clasice fără a avea implementate adresele G43 și G44

N60	T1	S12	M3	M6	LF
N65	G00	Z1500	D1	LF	Corecția D1 activă

- ❖ Activarea corecției fără a indica registrul (CNC Dialog 10)

N15	G0	Z300	Poziționare		
N20	G17	T1	Scula T1 activă, corecția de lungime pentru 1 este activă		
N25	G0	X20	Y16	Z2	S800 M3

Poziționare

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Determinarea valorii corecțiilor de lungime

- ❖ Cu dispozitive special de **prereglat**



- ❖ Cu dispozitive touch probe



- ❖ Pe mașina -unealtă

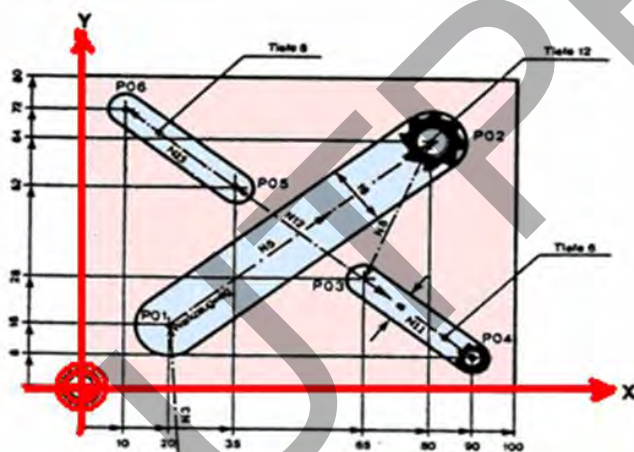
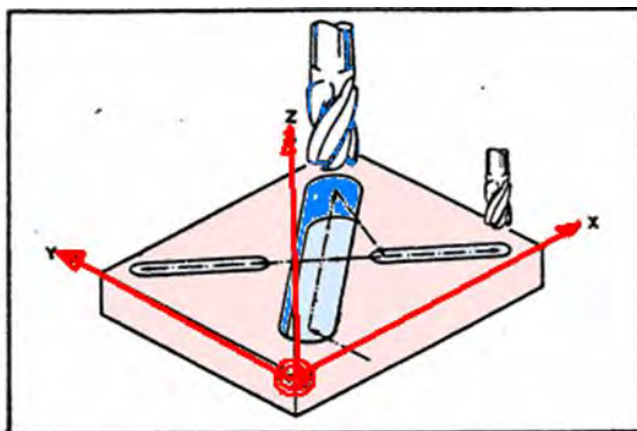
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice



```

% 46
N1 G0 Z100 ;      deplasarea de
                  siguranță
N2 G17 T1 ;      apelare scula 1,
                  corecția de lungime activă
N3 G0 X20 Y16 Z2 S800; punct P01
N4 G1 Z-12 F80;
N5 X80 Y64 F125;      Punct P02
N6 T0 ;              Anularea corecției de
                  lungime
N7 G0 Z100 S0;      Oprire AP
N8 G17 T2 ;          Apelare
                  scula2,corecția de lungime active
N9 G0 X65 Y28 Z2 S1600;
N10 Z-6 F60;
N11 X90 Y8 F100;
N12 G0 Z2;
N13 G0 X35 Y52;      Punctul P05
N14 Z-8 F60;         Frezare P05-P06
N15 X10 Y72 F100;
N16 T0;
N17 G0 Z100 S0 M30; Anulare CL
  
```

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Corecția traiectoriei programate a sculei

- Corecția de masă (paraxială)
- Corecția de rază

Asemănări: asigură deplasarea sculei după echidistantă programând centrul piesei

Deosebiri: corecția de masă trebuie „activată” bloc cu bloc (secvențial)

corecția de rază activată este valabilă pe tot conturul programat

## Corelația paraxială

Corecția paraxială (C.P.) se programează utilizând:

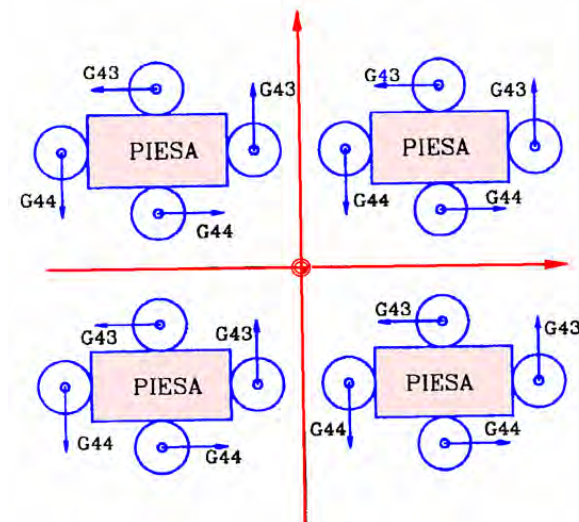
G43/G44 - adună scade valoarea corecției

D - registrul în care se introduce valoarea corecției

Are caracter model

*C.P. la E.C. de tipul N.C.*

Limitată la deplasări a sculei paralele cu axele programate X și Y



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## C.P. la C.N.C.

Programare:

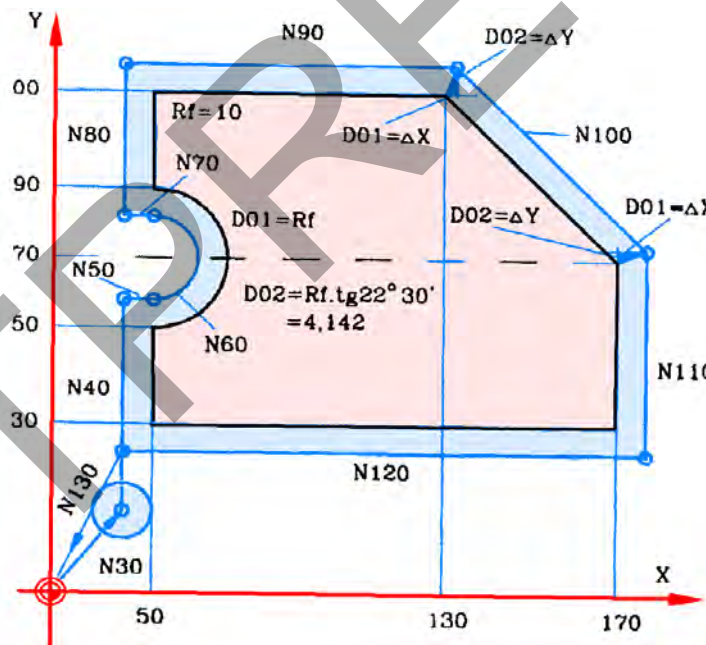
N20 G01 G43 X20000 D01 F50

Considerând valoarea memorată în D01=2500 , deplasarea efectivă X= 22500

Nu există restricții privind traiectoria sculei

Programare:

**G43/G44 D\_\_**



```
%20 LF
N10 G90 G17 S43 M3 LF
N20 G00 G44 D10 Z2000 LF
N30 G44 D01 X50000 Y10000
      LF
N40 G01 G43 D01 Y50000 F200
      LF
N50 G44 D00 X50000 LF
N60 G03 G44 D01 Y90000
      X50000 IO J20000 LF
N70 G01G44 D01 X50000 LF
.
.
N100 G43 D01 X170000 G43
      D02 Y70000 LF
.
.
N120 G44 D01 X50000 LF
N130 G44 D0 X0 D0 Y0 M30 LF
```

Obs. Corecția de lungime este un caz particular al corecției paraxiale.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Programare alternativă

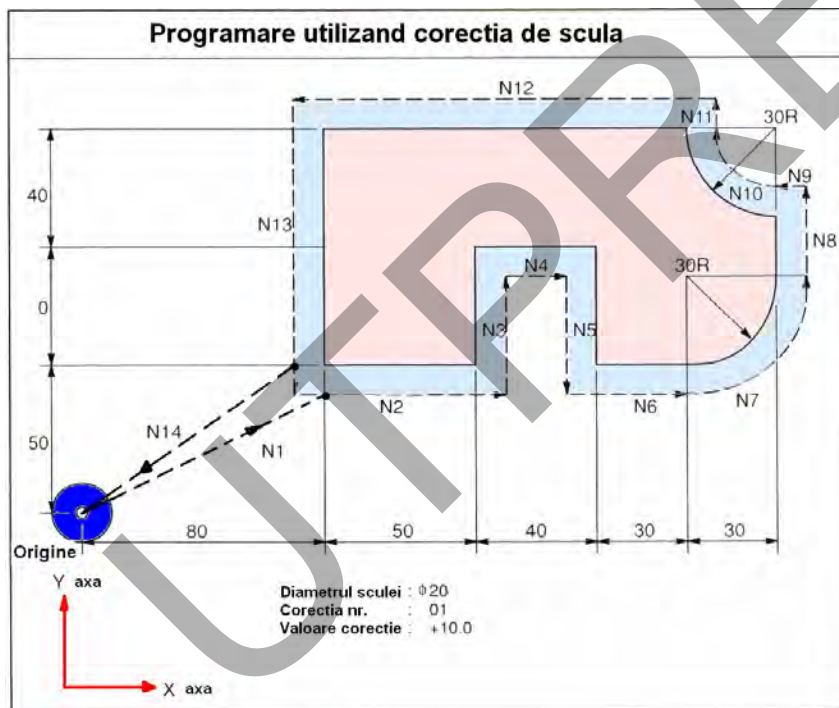
•funcții speciale:

**G45** axa D- crește

**G46** axa D- scade

**G47** axa D- crește dublu

**G48** axa D- scade dublu



Program

N1 G91 G46 G00 X80.0 Y50.0  
D01;

N2 G47 G01 X50.0 F120.0;

N3 Y40.0;

N4 G48 X40.0;

N5 Y-40.0;

N6 G45 X30.0;

N7 G45 G03 X30.0 Y30.0 J30.0;  
G45 G01 Y20.0;

N9 G46 X0;

N10 G46 G02 X-30.0 Y30.0  
J30.0;

N11 G45 G01 Y0;

N12 G47 X-120.0;

N13 G47 Y-80.0;

N14 G46 G00 X80.0 Y-50.0;



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

# Corecția de rază (CR)

Activarea corecției de rază:

**G41/G42**

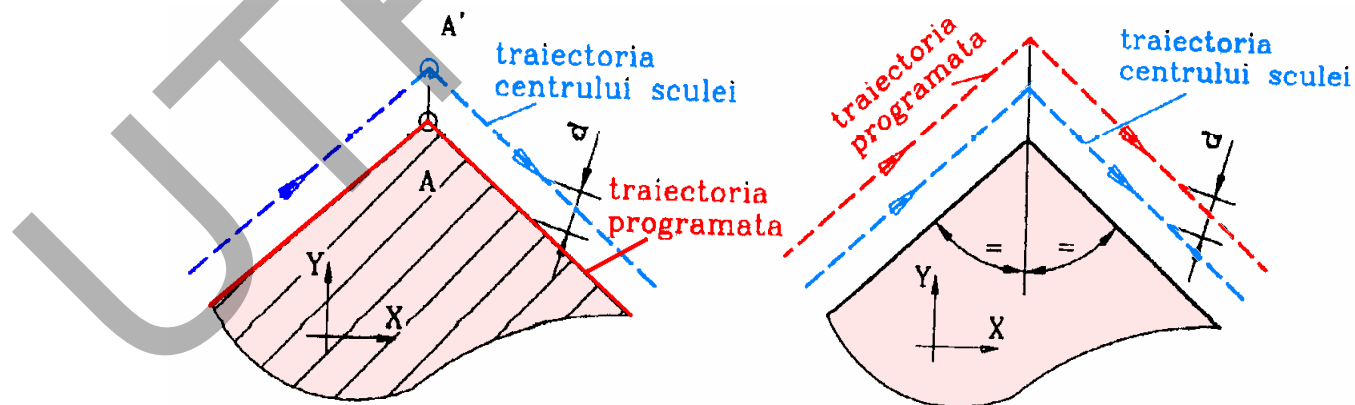
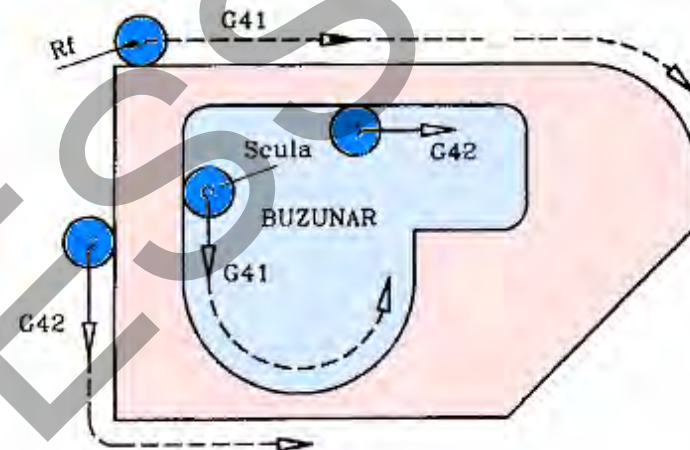
D \_\_ registrul în care se introduce  
valoarea corecției

Anularea corecției de rază

**G40/D0**

Corecția de rază :

Se utilizează pentru prelucrări cu periferia frezei. **Nu se utilizează la operații de burghiere, tarodare, filetare.**



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## ❖ De ce corecția de rază?

- Precizia suprafeței obținute la prelucrarea cu periferia frezei (profilare) depinde și de precizia frezei;
- Cât de apropiat este diametrul real de cel considerat în programare;
- Compensarea uzurii;
- Modificarea diametrului frezei; (ex: prin reascuțire, schimbare, etc.)
- Utilizarea unui singur program pentru degrosare și finisare;
- Compensarea unor abateri de la profilul obținut, în raport cu cel teoretic
- Etc.;

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Corecția de rază

NEGATIVĂ :  $R'_f - R_f < 0$

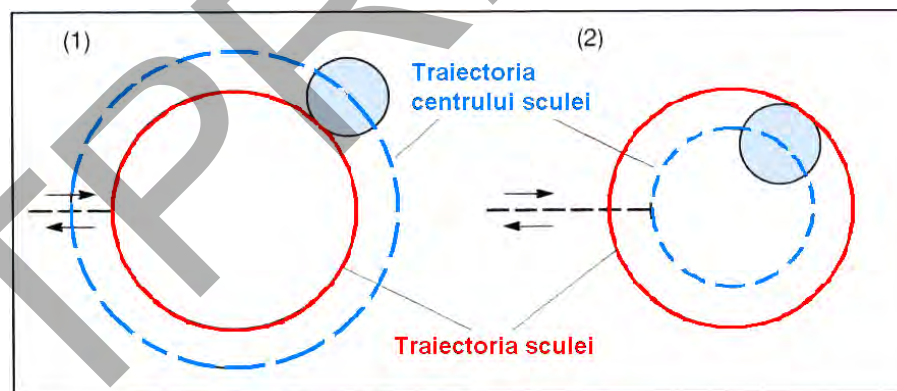
POZITIVĂ :  $R'_f - R_f > 0$

$R'_f$  - scula utilizată în prelucrare

$R_f$  - scula utilizată în programare

Nota: Posibila confuzie

La unele ECN de tipul CNC (FANUC Series Oi-MC) prin corecție negativă /pozitivă se înțelege semnul valorii corecției (nu este rezultatul unei scăderi).



Când traiectoria este programată ca în ( 1 ) și corecția este negativă, centrul sculei se va deplasa ca în ( 2 ), și invers.

Discuție: Prelucrarea de tip “moș - babă”

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

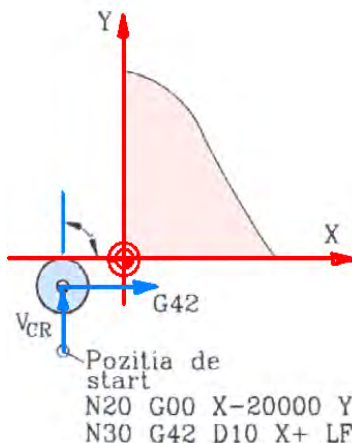
Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

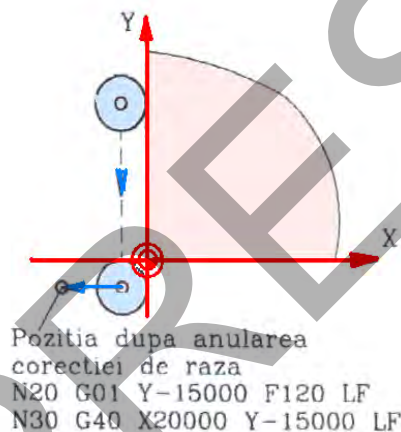
Cap.5.  
Informații  
tehnologice

## Corecția de rază la echipamentele N.C.

Activarea/anularea se face după o direcție perpendiculară pe direcția de deplasare.  
Vectorul C.R.



a)



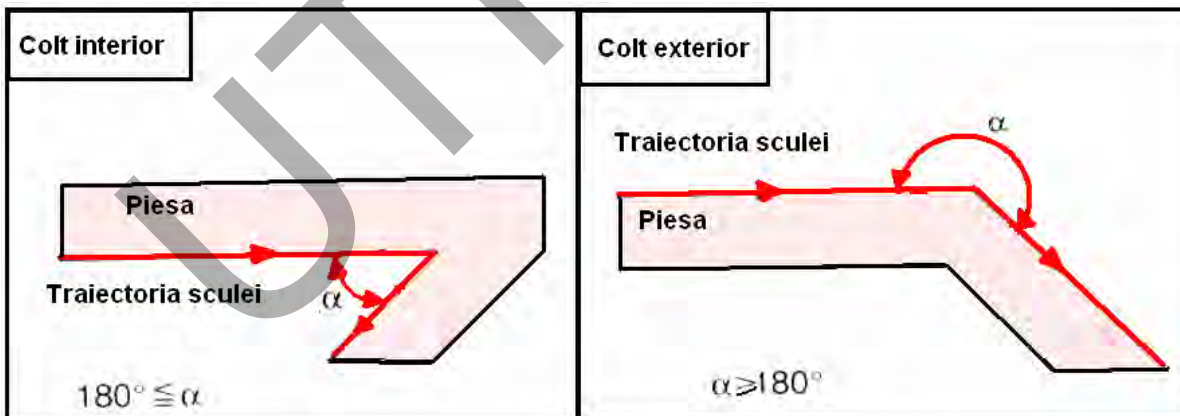
b)

Obs.: Unele ECN cer  
programarea G91 înainte de  
blocul în care s-a programat  
G41/G42 (VCR - deplasarea  
incrementală)

N20 G91 X - LF

N21 G42 D10 LF

N22 G90 LF



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

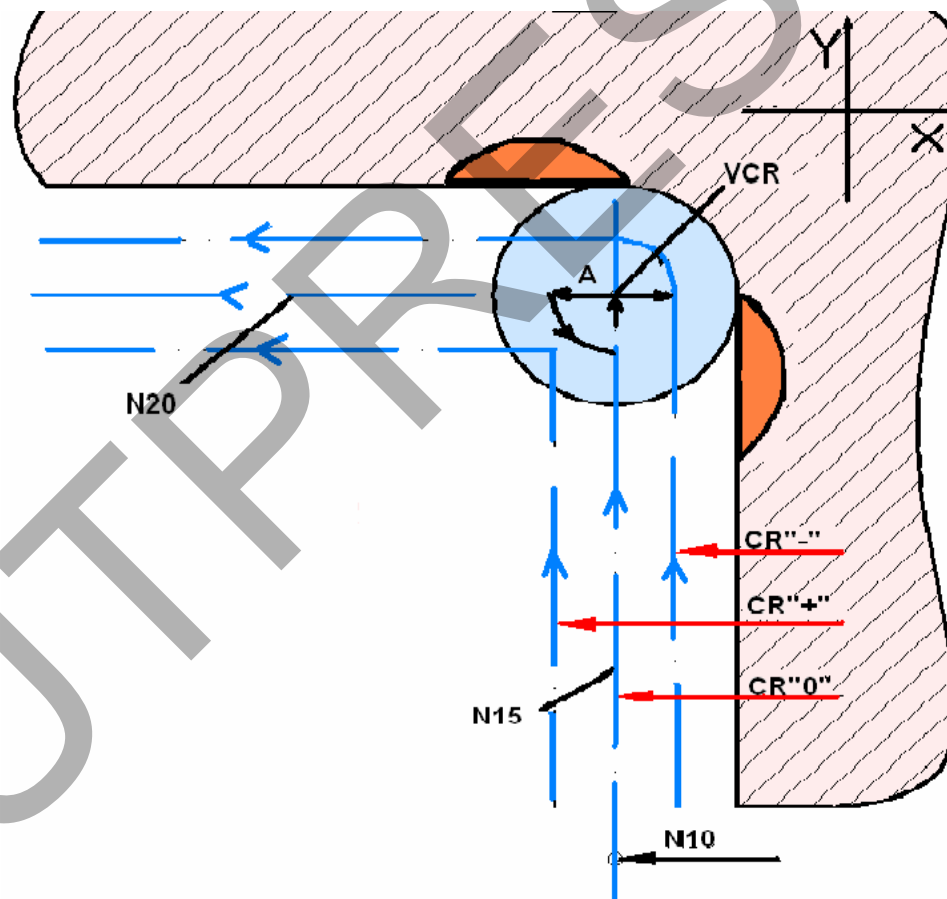
Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

O corecție de tipul:  $R'_f - R_f > 0$

Deformează colțurile interioare:



Cap.1.  
Introducere

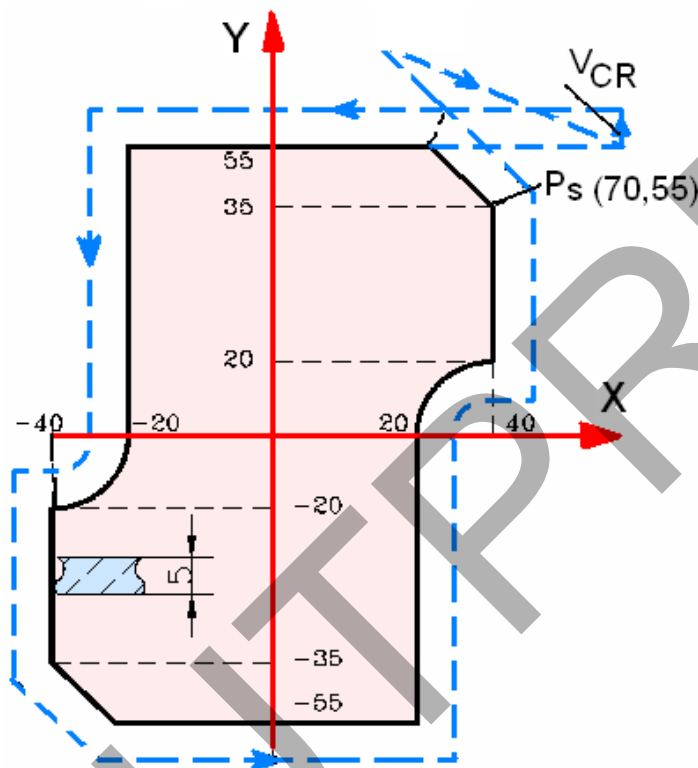
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Exemple de programare, echipamente NC (General Electric 550 MBD)



%LF

**N1 G90 LF**

**N5 G17 LF**

**N10 G04 X30000 S42 M03 LF**

(temporizare și pornire Ap)

**N20 G00 Z2000 D1 LF**

**N25 X 70000 Y15000 LF**

(poziționare în punctul de start, Ps)

**N30 Z-7000 LF**

**N35 G91 LF**

**N40 G42 X- D02 LF**

(programare corecție de rază  $R_f=10$  mm în D02)

**N45 G90**

**N50 G01 X-20000 F150 LF**

(programare contur piesă)

**N55 Y0 LF**

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

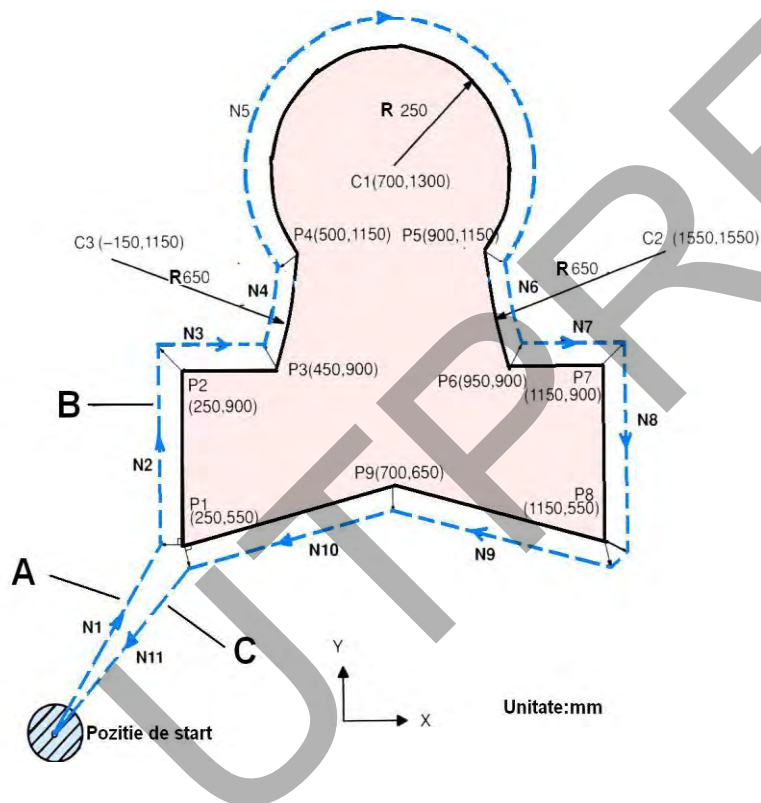
Cap.5.  
Informații  
tehnologice

## Corecția de rază (CR) la echipamente CNC

A.Traietoria sculei la activarea CR

B.Traietoria sculei în regimul OFFSET

C.Traietoria sculei la anularea CR.



G92 X0 Y0 Z0

N1G90 G17 G00 G41 D07

X250 Y550.0;

N2 G01Y900.0F150;

N3X450.0;

N4G03X500.0Y1150.0R650.

0;

N5G02X900.0R-250.0;

N6G03X950.0Y900.0R650.0;

N7G01X1150.0;

N8Y550.0;

N9X700.0Y650.0;

N10 X250.0Y550.0;

N11 G00 G40 X0 Y0

Specifica coordonate in sistem absolut. Scula este in pozitia(X0,Y0,Z0)

Activarea corectiei de raza, valoarea in D07

Deplasare din P1 in P2

Deplasare din P2 in P3

Deplasare din P3 in P4

Deplasare din P4 in P5

Deplasare din P5 in P6

Deplasare din P6 in P7

Deplasare din P7 in P8

Deplasare din P8 in P9

Deplasare din P9 in P1

Anulare CR

Revenire in pct. de start

**REMARCA:** Traietoria sculei, în cele trei regimuri poate fi programată prin diferite constante de mașină sau coduri G

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

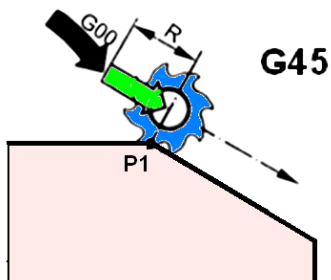
Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

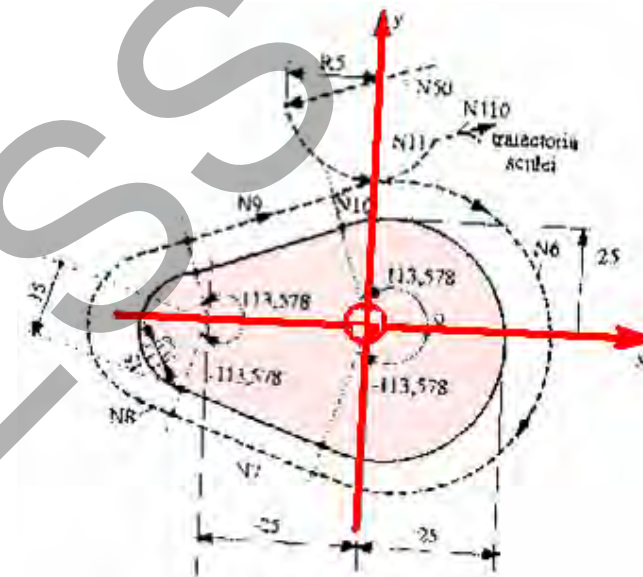
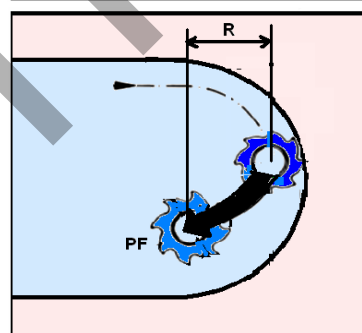
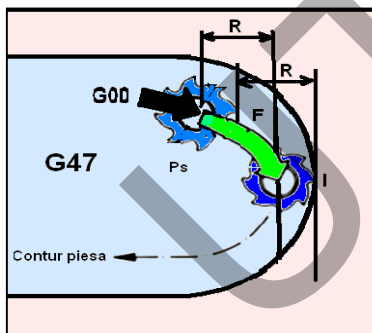
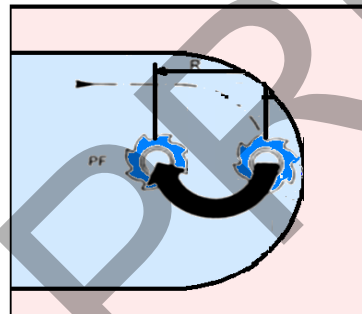
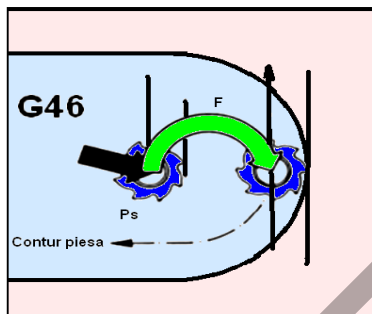
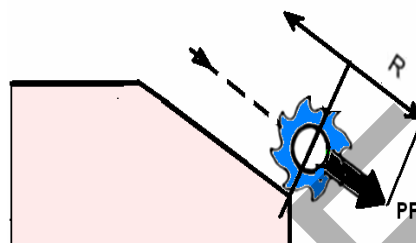
Cap.5.  
Informații  
tehnologice

## Trajectorii de activare/anulare a C.R. prin utilizarea unor funcții G

Activare



Anulare



% 50

N10 G00 Z100	LF
N20 G17 T01 M06	LF
N30 G00 Z2 S500	LF
N40 Z-17 F100	LF
N50 G00 G41 G47 R5 X0 Y25	LF
N60 G90 G09 G02 M72 W-113.578 I0 J0	LF
N70 G90 G09 G01 R15 W-113.578 I-25 J0	LF
N80 G90 G09 G02 M72 W113.578 I-25 J0	LF
N90 G90 G01 R25 W113.578 I0 J0	LF
N100 G90 G09 G02 M72 W90 I0 J0	LF
N110 G40 G47 R5	LF
N120 T0	LF
N130 G00 Z100 M30	LF



Cap.1.  
Introducere

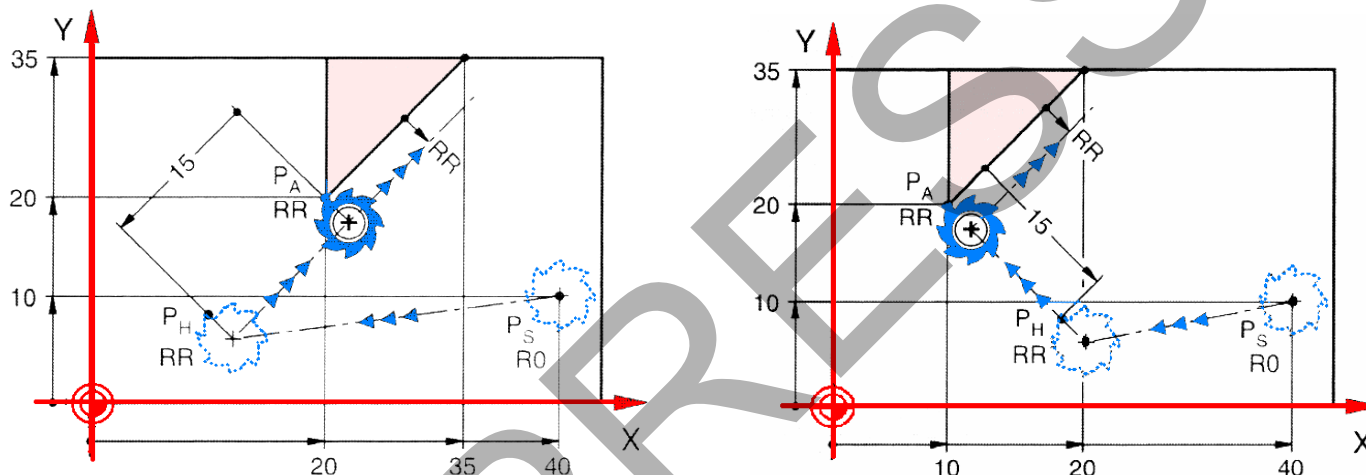
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Traiectorii de activare a C.R. prin funcții speciale



Unde:

$P_S$  punct de start

$P_H$  punct intermediar

$P_A$  primul punct de pe contur

$P_E$  ultimul punct de pe contur

$P_N$  punct final

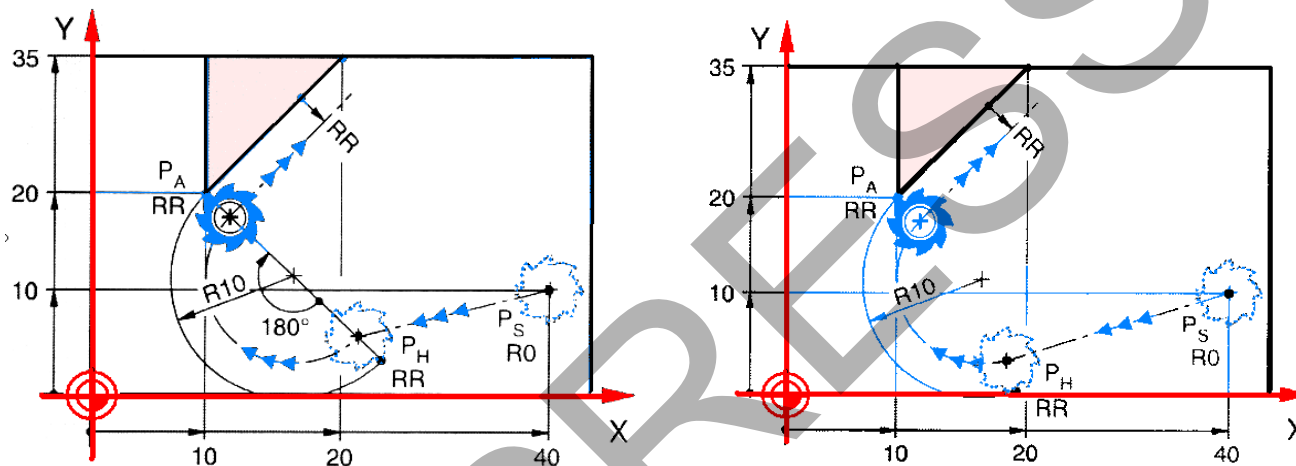
Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice



Unde:

$P_S$  punct de start

$P_H$  punct intermediar

$P_A$  primul punct de pe contur

$P_E$  ultimul punct de pe contur

$P_N$  punct final

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

- apropiere în linie dreaptă cu coordonare tangențială (APPR LT) (fig. 3.56. a)

```

7 L      X+40  Y10      R0  FMAX  M3
8 APPR  LT    X+10    Y+2  Z-15  LEN15  RR    F100
9 L      X+20  Y+35

```

- apropiere după o linie perpendiculară (APPR LN) (fig. 3.56. b)

```

7 L      X+40  Y+10    R0    FMAX  M3
8 APPR  LN    X+10    Y+25  Z-15  LEN15  RR    F100
9 L      X+20  Y+35

```

- apropiere după arc de cerc tangent la contur (APPR CT) (fig. 3.56. c)

```

7 L      X+40  Y+10    R0    FMAX  M3
8 APPR  CT    X+10    Y+20  Z-15  CCA18  R+10  RR    F100
9 L      X+20  Y+35

```

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

În versiunea de **programare ISO** a aceluiași echipament este implementată numai curba APPR CT, programabilă prin adresa G26.

<b>N50</b>	<b>G00</b>	<b>G40</b>	<b>G90</b>	<b>X+40</b>	<b>Y+10 *</b>
<b>N60</b>	<b>G01</b>	<b>G42</b>	<b>X+10</b>	<b>Y+20</b>	<b>F350 *</b>
<b>N70</b>	<b>G26</b>	<b>R6 *</b>			

Depărtarea de contur se programează prin adresa G27.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

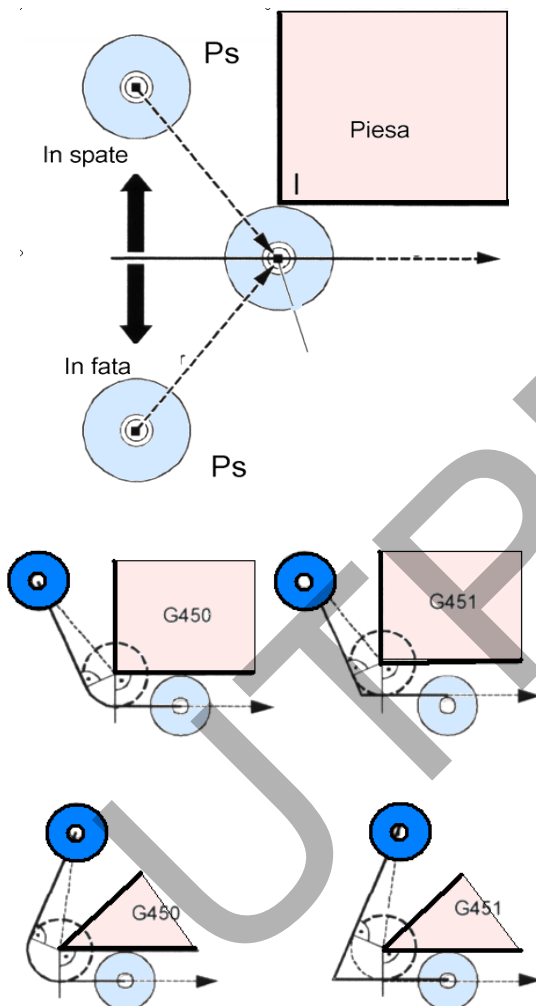
Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

## Comanda: KONT G41/G42

Traectoria sculei depinde de poziția relativă a punctului I în raport de punctul de start Ps



%	N_EXEMPLU_MPF							
N10	G0	Z100						
N20	G17	T1	M6					Schimbare sculă
N30	G0	X0	Y0	Z1	M3	S800	D1	Apelare valori offset sculă, activare corecție de lungime
N40	G01	Z-7	F500					
N50	G41	KNOT	G450	X20	Y20			Apelare corecție de rază, scula pe stânga
N60	Y40							
N70	X40	Y70						
N80	X80	Y50						
N90	Y20							
N100	X20							
N110	G40	G0	Z100	M30				Anulare corecție de rază, retragere sculă

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

## Cap. 6. CICLURI FIXE PROGRAMAREA FLEXIBILĂ

O secvență de operații inițiate prin programarea unei funcții G. Ciclurile fixe acționează ca un short cuts care simplifică programul.

Operație: Deplasări ale sculei

Fac posibilă realizarea mai ușoară a unui program.

Cu ajutorul ciclurilor fixe este posibilă programarea unei operații de prelucrare (ex. Burghiarea într-un singur bloc specificându-se o anumită funcție G).

Cap.1.  
Introducere

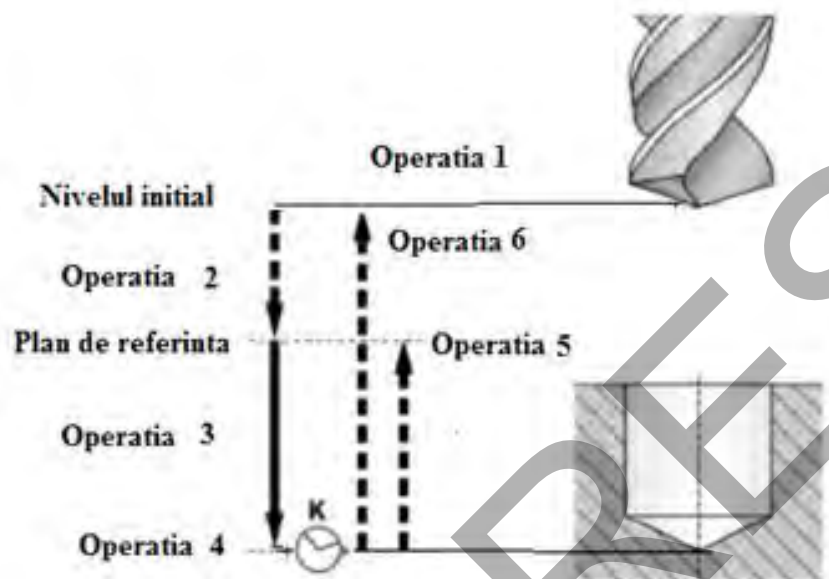
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe



Operația 1: Poziționare pe axele X, Y

Operația 2: Deplasare rapidă în planul de referință R

Operația 3: Prelucrarea găurii

Operația 4: Operații la fundul găurii

Operația 5: Retragere rapidă în R

Operația 6: Retragere rapidă în punctul inițial

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

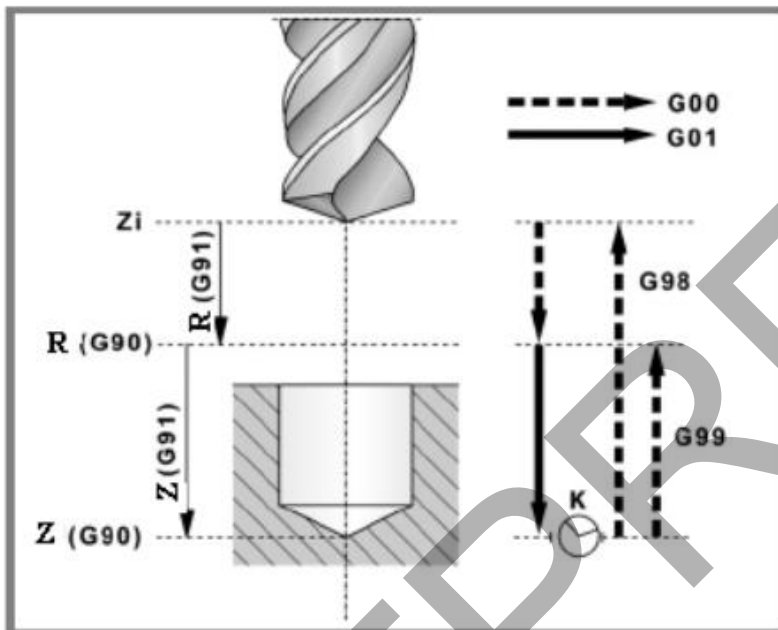
Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

## Definiri diferite ale parametrilor ciclului



General

G81 ciclu fix

R planul de referință

Z adâncimea găurii

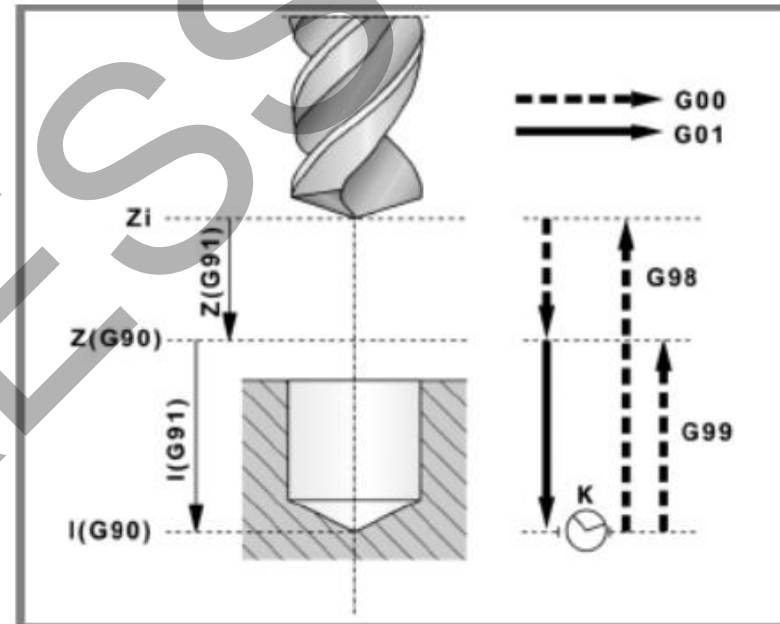
Zi poziționarea inițială

G99 revenire în referință

G98 revenire în poziția

inițială

K oprire la cotă



Fagor

G 81 ciclu fix

Z planul de referință

I adâncimea găurii

Zi poziționarea inițială



# Modificare parametrii ciclu

Cap.1.  
Introducere

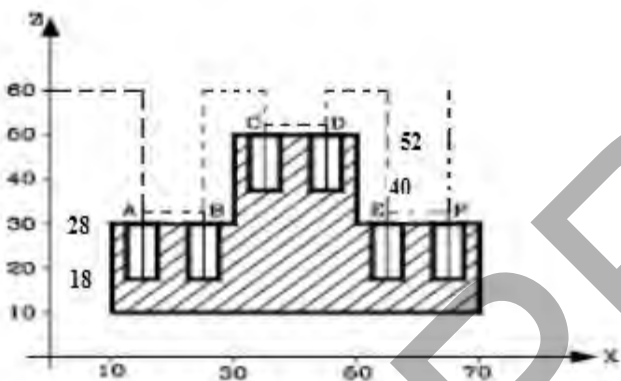
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe



Comanda	Explicații
T1 M6;	
G00 G90 X0 Y0 Z60 F1000;	Punct de start
G99 G91 X15 Y25 G81 Z-28 I-14;	Ciclu pt. gaura A
G98 G90 X25;	Ciclu pt. gaura B
V.C. Z=25 V.C. I=40;	Valorile modificate pt. Z si I
G99 X35;	Ciclu pt. gaura C
G98 X45;	Ciclu pt. gaura D
V.C. Z=32 V.C. I=18;	
G99 X55;	Ciclu pt. gaura E
G98 X65;	Ciclu pt. gaura F

Modificarea (echipament FAGOR) parametrilor se realizează prin variabile pentru parametru V.C.A, V.C.R, etc.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

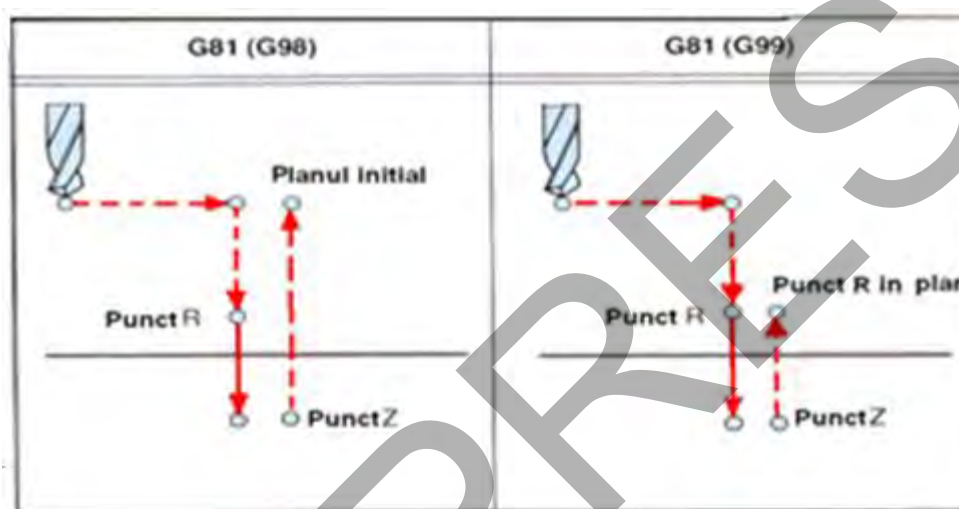
Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

# G81 Găurire (normală)

G81 X...Y...Z...R...F...K...



X, Y Poziționare gaură  
Z Distanța între punctul R și fundul  
găurii  
R Distanța de la nivelul inițial la  
punctul R  
F Viteza de avans  
K Număr de repetări

M3 S2000

G90 G99 G81 X300. Y-250. Z-150. R-100. F120;  
Poziționare gaura 1 și revenire în R  
Y-550;

....

G98 Y-750.; Poziționare gaură... revenire în  
punctul inițial

G80 G28 G91 X0 Y0 Z0;  
M5;

Cap.1.  
Introducere

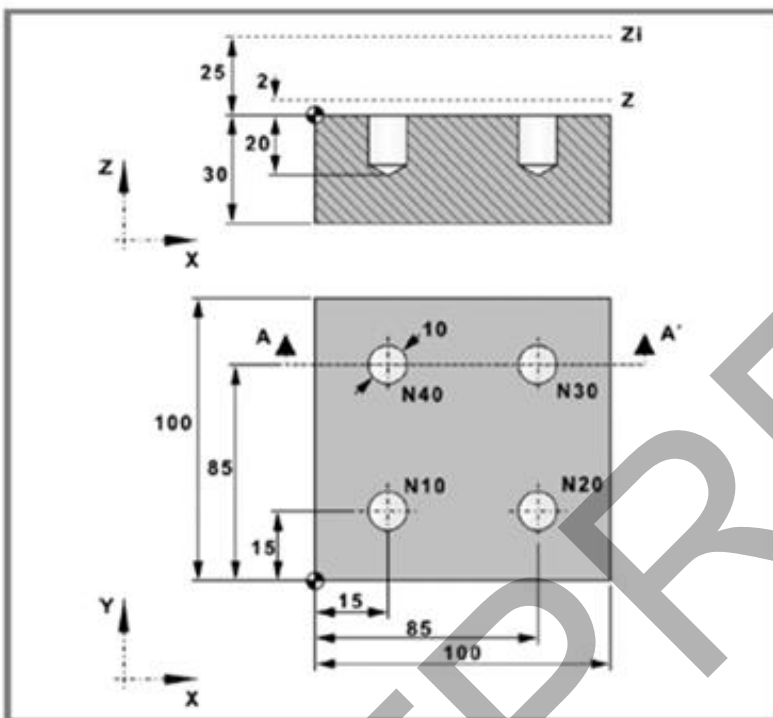
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe



## FANUC

### G90

```
G0 G90 X15. Y15.0 X0 Y0 Z25. S2000 M3
F200;
G99 G81 X15. Y15. Z-20. R2;
X85.;
Y85.;
G98 X15.;
M30;
```

## FAGOR

Exemplu de programare

### G90

```
G0 G90 X0 Y0 Z25 S1000 M3 F200
N10 G99 X15 Y15 G81 Z2 I-20
N20 X85
N30 Y85
N40 G98 X15
M30
```

### G91

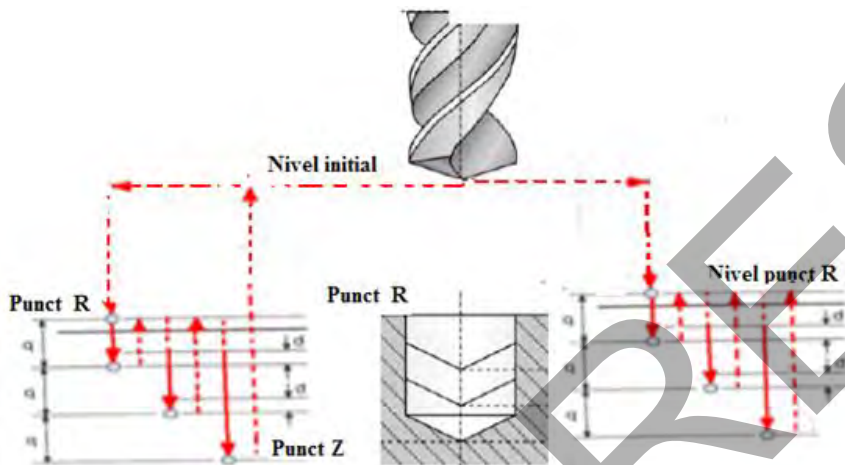
```
G0 G90 X0 Y0 Z25 S1000 M3 F200
N10 G99 G91 X15 Y15 G81 Z-23 I-22
N20 X70
N30 Y70
N40 G98 X-70
M30
```

### G91

```
G0 G91 X15. Y15.0 X0 Y0 Z25. S2000
M3 F200;
G99 G81 X15. Y15. Z-22. R-23.;
X70.;
Y70.;
G98 X-70.;
M30;
```

# G83 (Găurire adâncă)

G83 X...Y...R...F...K...;



## FANUC

X, Y Poziționare gaură

Z Distanța între punctul R și fundul găurii

R Distanța de la nivelul inițial la punctul R

Q Adâncimea de pătrundere/ trecere

Program	Explicație
S2000 M03	
G90 G99 G83 X300. Y-250. Z-150. R-100 Q15. F120.;	Poziționare la prima gaură, cu revenirea în R
Y-550.	Poziționare la gaura a 2-a
X1000.	Poziționare la gaura a 3-a
G98 Y-750.	Poziționare la gaura a 4-a cu revenire în punctul inițial
G80 G28 G91 X0 Y0 Z0	Revenire în punctul de referință
M5	Oprire AP

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

# G84 Tarodare

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

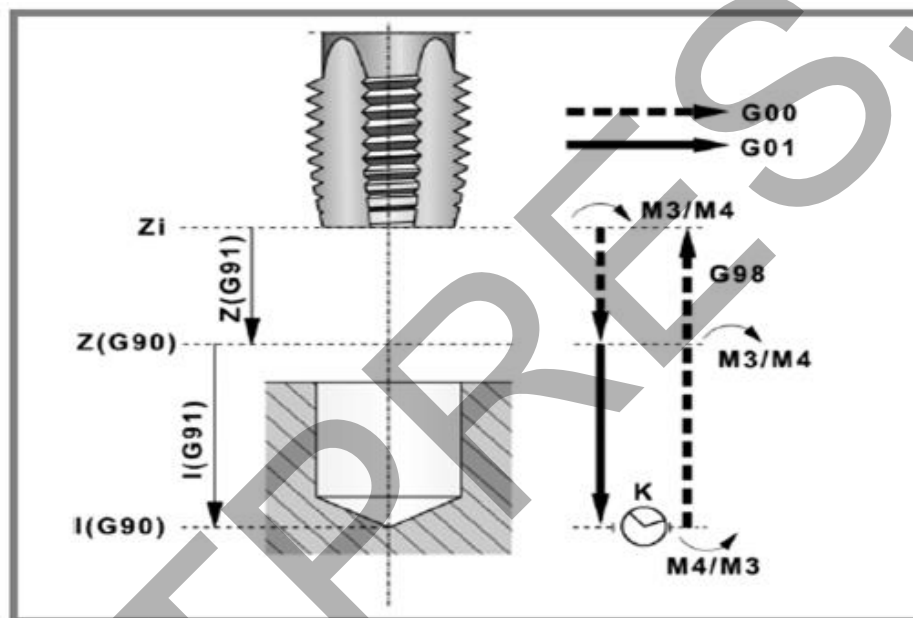
Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

G84 Z...I...K...R...J... Fagor



Z Planul de referință, se raportează G90 - originea piesei  
G91 – punctul de start

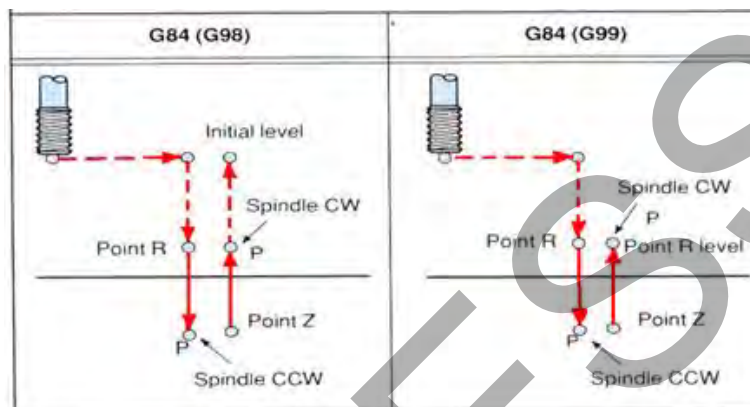
I Adâncimea filetului

K Stop

R Tip de tarod R0- normal R1- rigid

J Factor pentru modificarea vitezei de retragere

## Ciclul de tarodare (G84)



G84 X\_\_Y\_\_Z\_\_R\_\_P\_\_F\_\_K\_\_ ;  
 X\_\_Y\_\_ : Poziția găurii  
 Z\_\_ : Distanța de la R la fundul găurii  
 R\_\_ : Distanța de la planul Z=0 la planul de referință  
 P\_\_ : Timpul de oprire  
 F\_\_ : Viteza de avans  
 K\_\_ : Număr de repetări (dacă este necesar)

M3 S100;	Pornire AP
G90 G99 G84 X300.Y250.Z-150.R-120.P300 F120.;	Poziționare la gaura 1 cu revenire în R
Y550.;	Găurire gaura 2 cu revenire în R
Y750.;	Găurire gaura 3 cu revenire în R
X1000.;	Găurire gaura 4 cu revenire în R
Y550.;	Găurire gaura 5 cu revenire în R
G98 Y750.;	Găurire gaura 6 cu revenire în punctul inițial
G80 G28 G91 X0 Y0 Z0;	Deplasare în punctul de referință
M5;	Oprire AP

Cap.1.  
Introducere

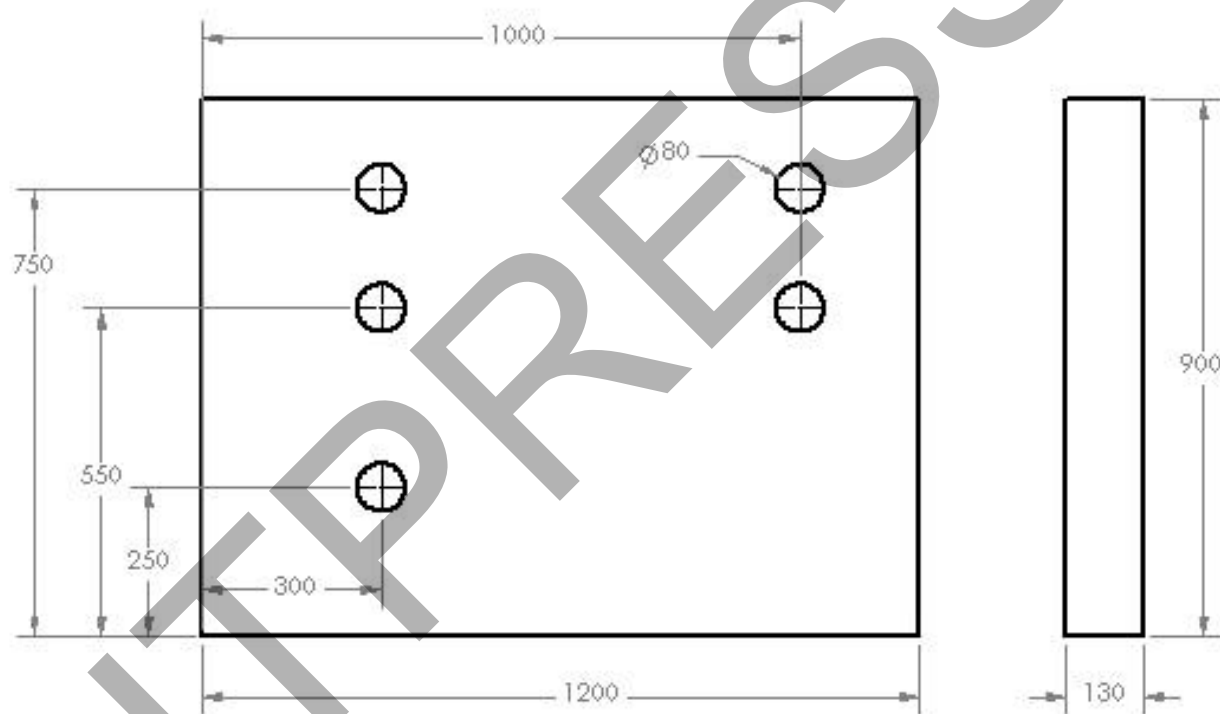
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

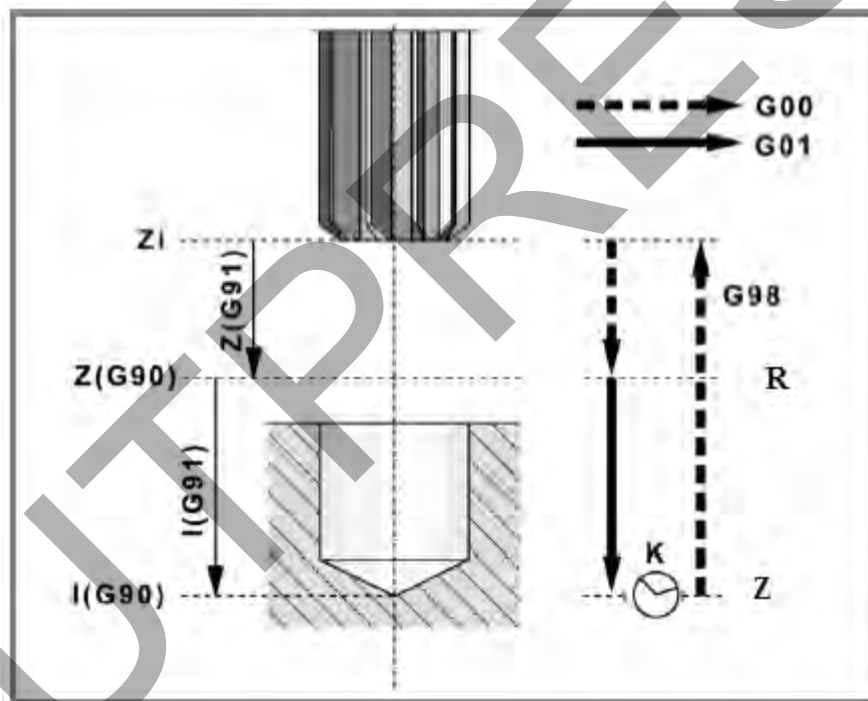
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

# G85 Alezare

FAGOR G85 Z...I...K...

FANUC G85 X...Y...R...F...K...;





Cap.1.  
Introducere

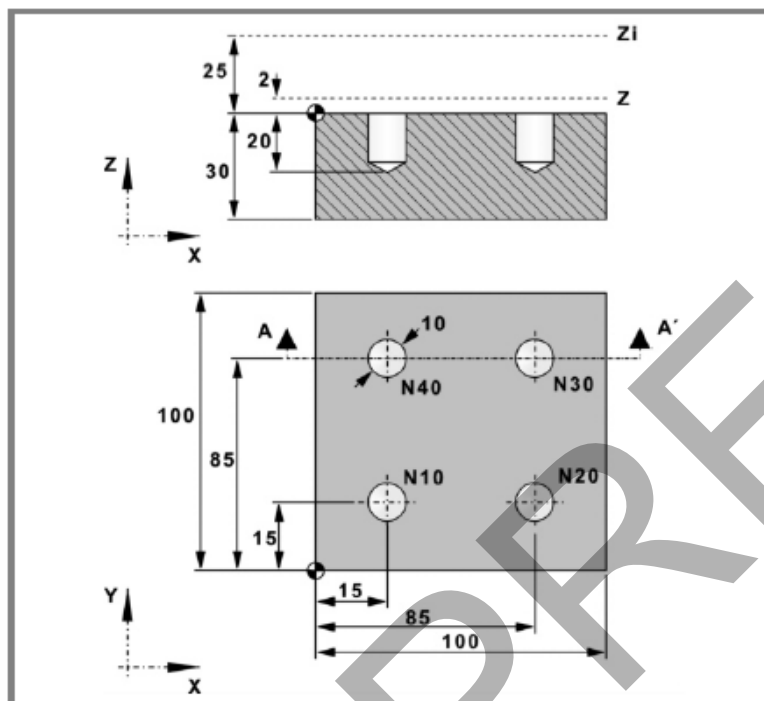
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe



## FAGOR

### G90

```
G0 G90 X0 Y0 Z25 S1000 M3 F200
G99 X15 Y15 G85 Z2 I-20
X85
Y85
G98 X15
M30
```

### G91

```
G0 G90 X0 Y0 Z25 S1000 M3 F200
G99 G91 X15 Y15 G85 Z-23 I-22
X70
Y70
G98 X-70
M30
```

## FANUC

### G90

```
G0 G90 X0 Y0 Z25 S1000 M3 F200;
G99 X15 Y15 G85 R2. Z-20.;
X85.;
Y85.;
G98 X15.;
M30
```

### G91

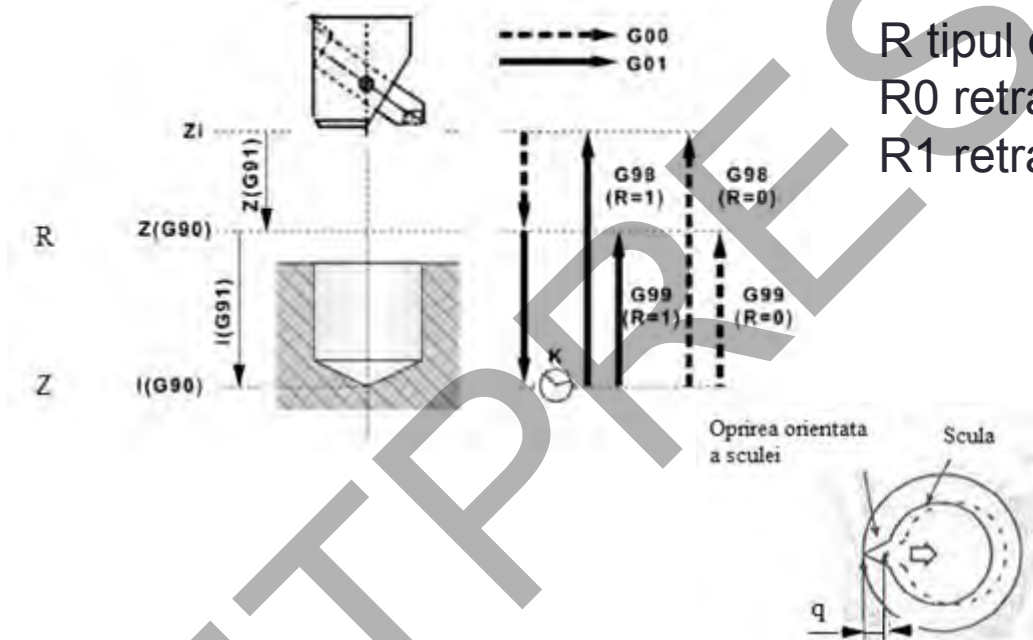
```
G0 G90 X0 Y0 Z25 S1000 M3 F200;
G99 G91 X15 Y15 G85 Z-23 I-22;
X70;
Y70;
G98 X-70.;
M30
```

# G86 Alezare

G86 Z...I...K...R...;  
G86 X...Y...Z...R...F...K...;

## FANUC

R tipul de retragere  
R0 retragere rapidă  
R1 retragere cu avans de lucru



G87 X...Y...Z...R...Q...P...F...K...;

Q retragere la fundul găurii  
P stop

Cap.1.  
Introducere

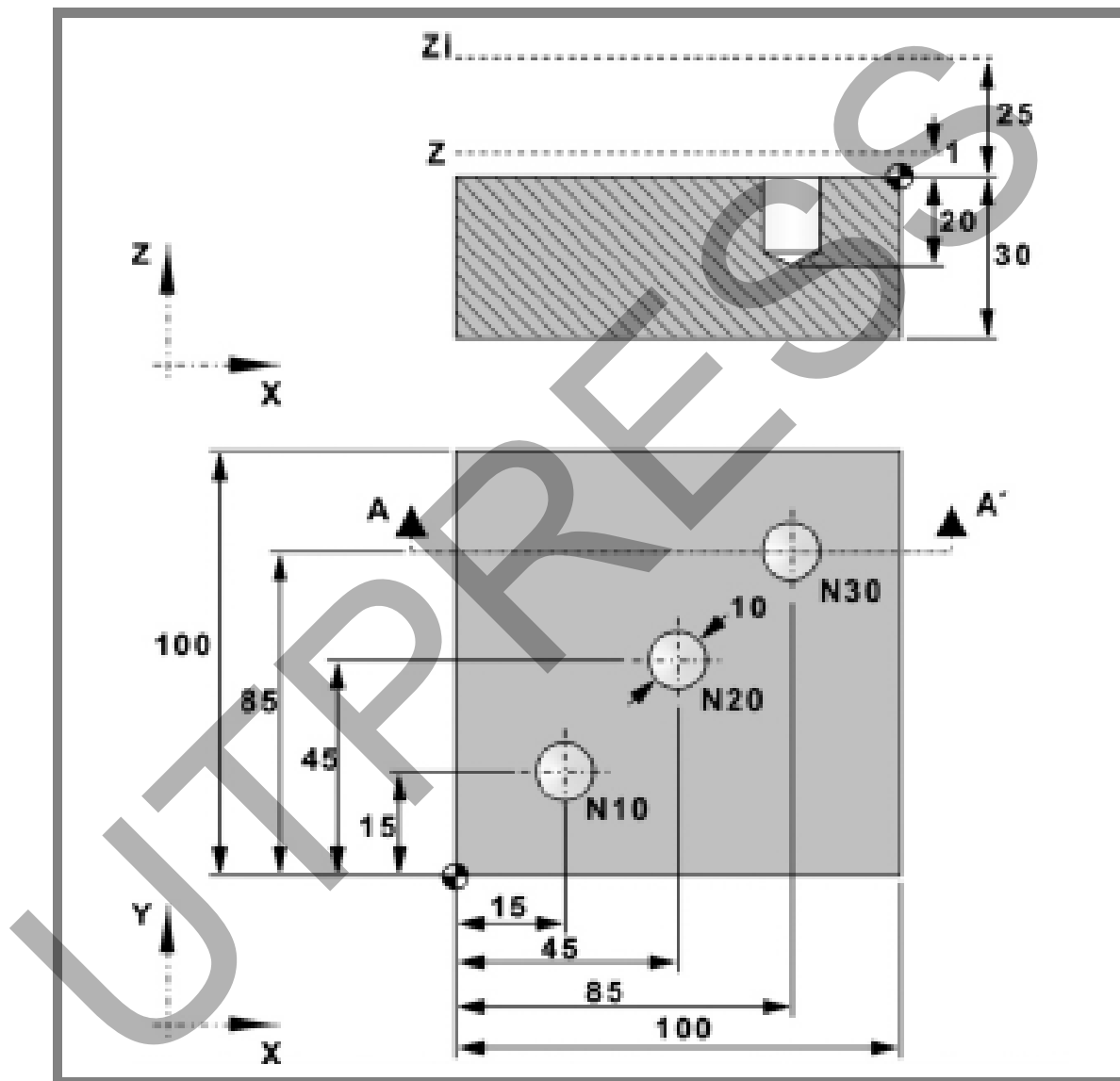
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

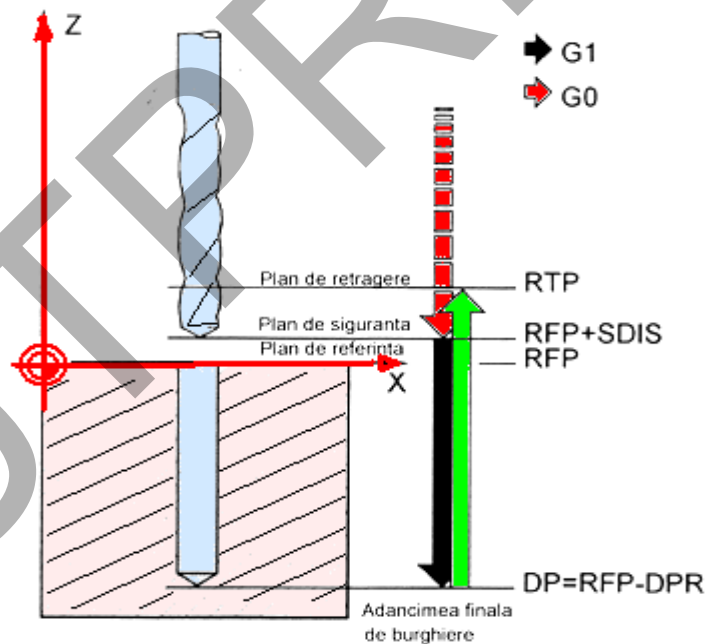
# Programarea ciclurilor fixe pentru găurire

(echipament SINUMETRIK 840D)

Modul de programare depinde de ECN

Exemplele prezentate anterior sunt compatibile cu echipamentele numerice din familia FANUC.

Echipamente numerice din familia SINUMERIK (840 D)



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

## Programare:

### Ciclul de găurire – CYCLE 81

**CYCLE 81 (RTP, RFP,SDIS, DP, DPR)**

#### În care:

- RTP** -planul de retragere sculă (valoare absolută)
- RFP** -planul de referință (valoare absolută)
- SDIS** -distanța de siguranță (fără semn)
- DP** -adâncimea finală de burghiere (valoare absolută)
- DPR** -adâncimea finală de burghiere relativă la planul de referință (fără semn)



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

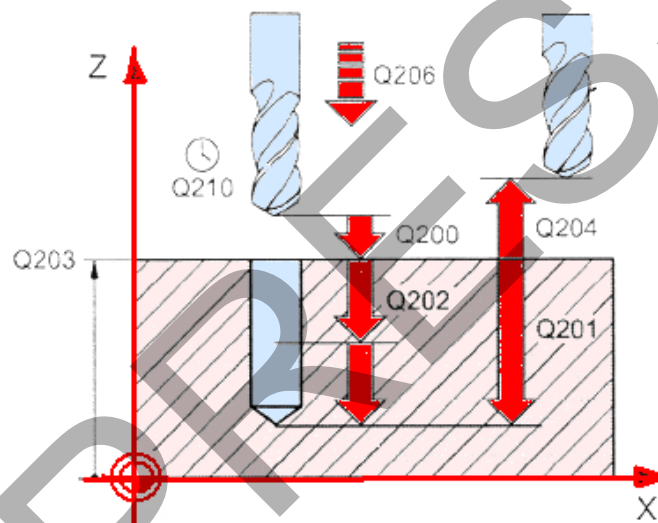
Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

## Un ciclu asemănător programat pe echipamentul Heidenhain (TNC 430)



În programare se utilizează parametrii Q

N70 G200 Q200=2 Q201=-20.....

În care G200- ciclul 200

....

....

Q210- oprire avans

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

# Cicluri de frezare

Echipamentele CNC moderne au implementate o mulțime de cicluri pentru frezare:

- frezarea filetelor,
- a găurilor alungite, dispuse pe un cerc,
- a buzunarelor dispuse după un cerc,
- a buzunarelor sub forma unor sectoare de cerc,
- a buzunarelor individuale rectangulare,
- a celor circulare, a suprafețelor plane,
- a suprafețelor de tip treaptă,
- a buzunarelor prevăzute cu insule,
- a unor suprafețe înclinate,
- a prelucrării unor suprafețe complexe compuse din mai multe entități plasate interior sau exterior (cicluri SLM) sau înlănțuite.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

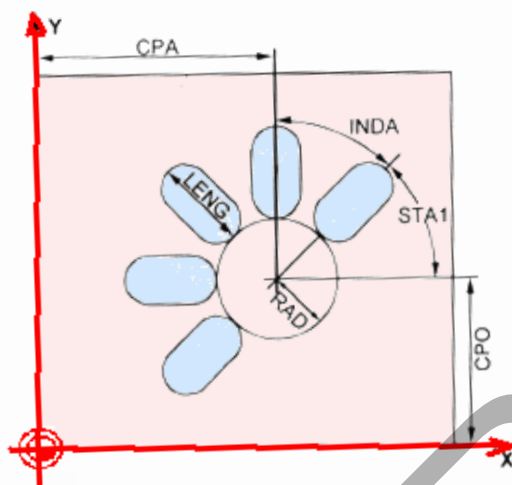
Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Găuri alungite dispuse pe un cerc, ciclu LONGHOLE ( fig.)(Sinumerik 840 D)



Programare

*LONGHOLE(RTP, RFP, SDIS, DP, DPR, NUM, LENG, CPA, CPO, RAD, STA1, INDA, FFD, FFM, MID)*

Semnificația parametrilor noi este următoarea:

- LENG* -lungimea găurii alungite (fără semn)
- FFD* -avansul de pătrundere
- FFM* -avansul pentru prelucrarea plană a găurii
- MID* -adâncimea de pătrundere maximă la o trecere

Semnificația celorlalți parametrii (primii cinci au fost specificați pentru ciclurile de găurire)

Cap.1.  
Introducere

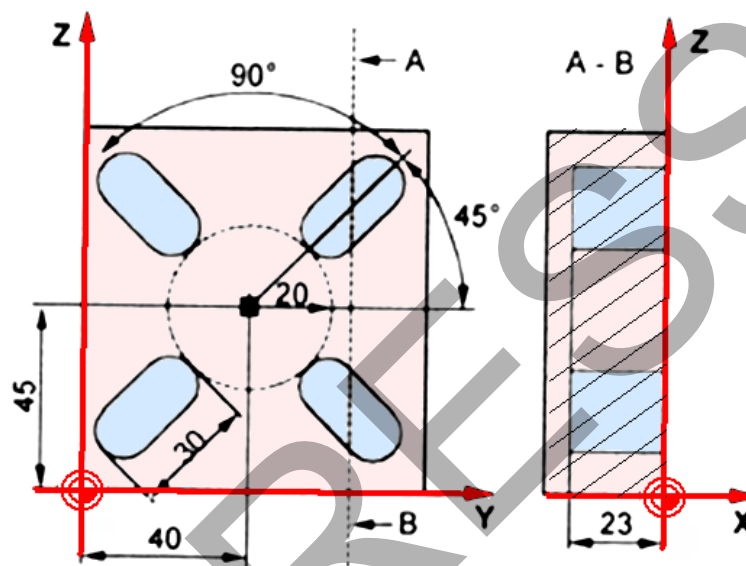
Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe



```

N10 G19 G90 S600 M3
T10 D1 M6
N20 G0 Y50 Z25 X5
N30 LONGHOLE(5, 0, 1, , 23, 4, 30, 40,
45, 20, 45, 90, 100, 320, 6)
N40 M30
  
```

- valori tehnologice, schimbare scula, activare CL,
- oprire orientată AP
- deplasare în poziția de start
- apelare ciclu
- sfârșit program

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

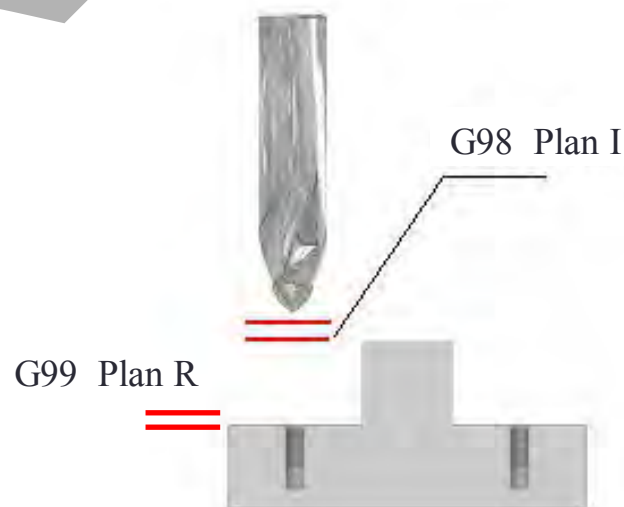
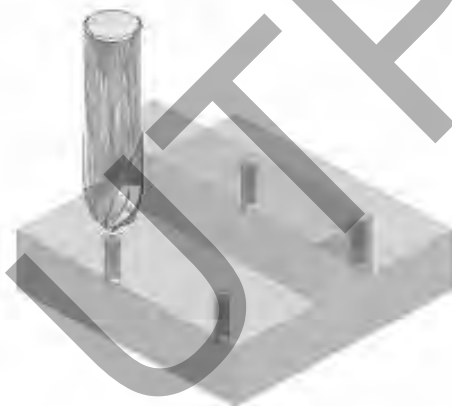
Cap.6. Cicluri  
fixe

## Piese cu prag

Când se prelucrează astfel de găuri se recomandă utilizarea, ca punct de start în ciclul, a două planuri diferite.

Echipamentele CNC admit programarea

- ❖ unui plan inițial (I) și revenirea în el prin G98
- ❖ planul de siguranță (R) și revenirea în el prin programarea funcției G99.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

## Planul inițial I.



Planul I, este situat la distanța de 5.0 mm deasupra pragului, locație în care urmează să fie poziționată scula înainte de inițierea unui ciclu fix. Pentru exemplul nostru, poziționarea se programează astfel:

G0	G90	G57	X50.0	Y50.0
G43	H1	Z5.0		M8

Revenirea în planul I se programează prin G98 (Fanuc)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

## Planul R



Planul R, în care se aduce scula înainte de inițierea ciclului de găurire, este situat la o distanță de siguranță (ex. 5 mm) de suprafață piesei. Pentru exemplu prezentat R se găsește la Z-10. Retragerea rapidă a sculei, la revenire, în planul R se realizează prin programarea codului G99 (Fanuc).

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

## Ex.Ciclul fix găurire

%						
012						
N001	G92	X0	Y0	Z0;		Setare la poziția de referință
N002	G90	G0	Z250.0	T11	M6;	Schimbare sculă
N003	G43	Z5.0	H11	;		Nivelul inițial, l, corecția de lungime
N004	S250	M3;				Pornire AP
N005	G99	G81	X50.0	Y50.0;		Poziționare gaura 1 și găurire
	Z-35.0	R-10.0	F80;			
N006	X150.0;					Poziție în gaura #2, revenire în R
N007	G98	Y250.0 ;				Poziție în gaura #3, revenire în I
N008	G98	X50.0 ;				Poziție în gaura #4, revenire în R
N009	G00	X0	Y0	M05 ;		Revenire în poz. de referință.
N010	G49	Z250.0	T21	M6 ;		Schimbare sculă, anulare CL
N011	G43	Z5.0	H21			CL activată, revenire în planul I (z=5.0 mm)
N012	G85	G99	X50.0	Y150.0		Alezare gaura #5
	Z-48.0	R5.0	F50;			
N013	X150.0 ;					Alezare gaura #6
N014	G28	X0	Y0	M5 ;		Retragere în punctul de referință .
						Stop AP.
N015	G49	Z0 ;				Anulare CL
N016	M0	;				
%						

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

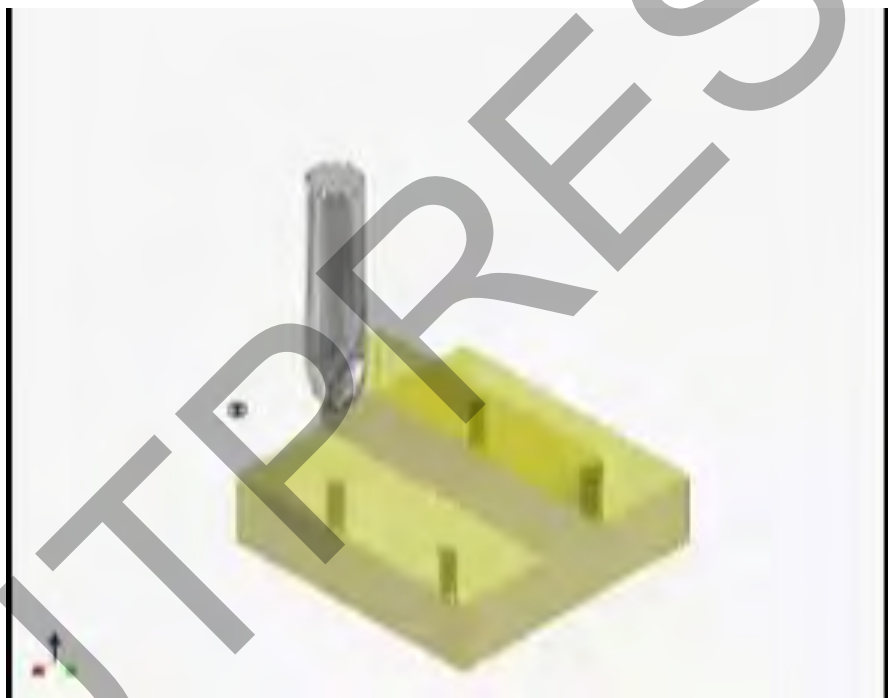
Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

## Animație



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Cap. 7. SUBPROGRAME

De ce subprograme?

Pentru a simplifica programul de prelucrare când prelucrarea implică o anumită secvență fixă de acțiuni sau o structură ce se repetă frecvent.

Apelarea subprogramului:

- în programul principal
- în subprograme (înlănțuire de maxim 5)

Un subprogram se identifică prin:

- prin numele (adresă, cifră...)
- prin instrucțiunea de sfârșit (M17, M99, ...)

Numărul de repetări, definit prin modul de apelare.

Adresele din cadrul subprogramelor pot fi urmate de:

- cifre;
- parametri;



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

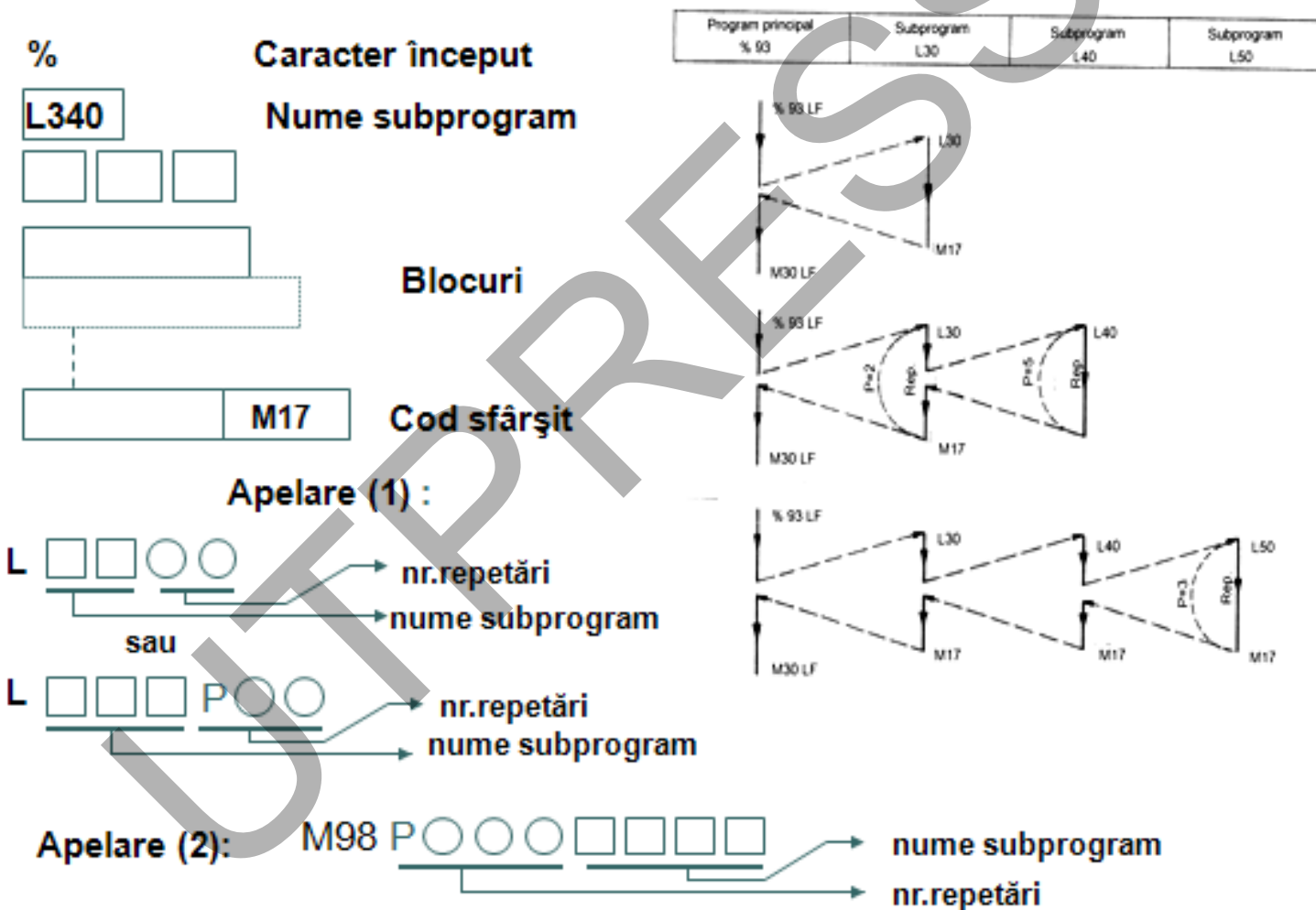
Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Configurația unui subprogram



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Varianta simplă de programare

(T01 burghiu centruire)

N1 G21

N2 G17 G40 G80

N3 G90 G54 G00 X7.0 Y7.0 S1200 M03 T02

N4 G43 Z25.0 H01 M08

N5 G99 G82 R2.5 Z-3.4 P200 F200.0

(H1)

N6 X39.0

(H2)

N7 Y45.0

(H3)

N8 X7.0

(H4)

N9 X23.0 Y26.0

(H5)

N10 G80 G00 Z25.0 M09

N11 G28 Z25.0 M05

N12 M01

(T02 burghiu 5mm)

N13 T02

N14 M06

N15 G90 G54 G00 X7.0 Y7.0 S950 M03 T03

N16 G43 Z25.0 H02 M08

N17 G99 G81 R2.5 Z-10.5 F300.0

(H1)

N18 X39.0

(H2)

N19 Y45.0

(H3)

N20 X7.0

(H4)

N21 X23.0 Y26.0

(H5)

N22 G80 G00 Z25.0 M09

N23 G28 Z25.0 M05

N24 M01

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

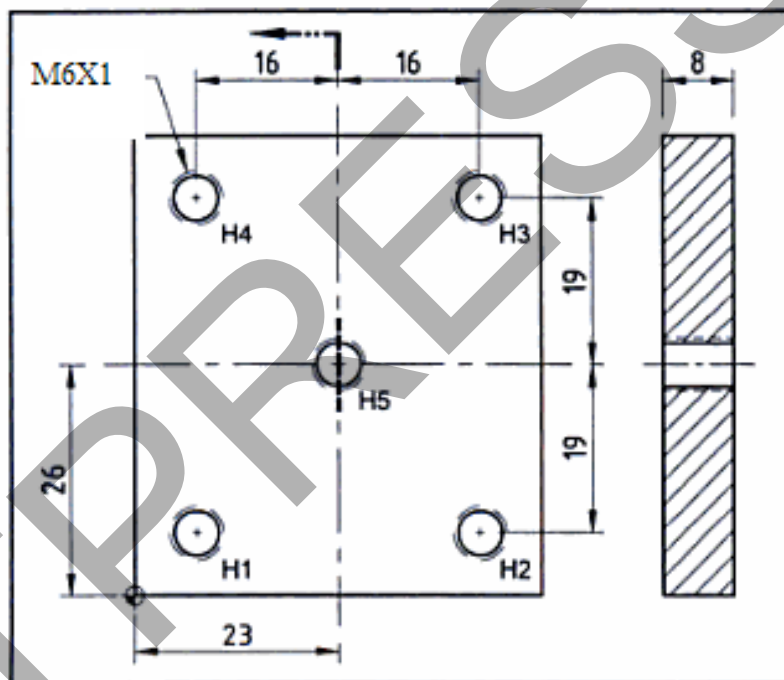
Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme



Cap.1.  
Introducere

T03 tarod M6X1

Cap.2.  
Arhitectura  
unui program

N25 T03

N26 M06

Cap.3.  
Limbaaj ISO

N27 G90 G54 G00 X7.0 Y7.0 S600 M03 T01

N28 G43 Z25.0 H03 M08

Cap.4.  
Programarea  
deplasărilor

N29 G99 G84 R5.0 Z-11.0 F600.0 (H1)

N30 X39.0 (H2)

Cap.5.  
Informații  
tehnologice

N31 Y45.0 (H3)

Cap.6. Cicluri  
fixe

N32 X7.0 (H4)

N33 X23.0 Y26.0 (H5)

Cap.7.  
Subprograme

N34 G80 G00 Z25.0 M09

N35 G28 Z25.0 M05

N36 G28 X23.0 Y26.0

N37 M30

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Subprogram varianta 1

T01 (burghiu centruire montat)

N2 G17 G40 G80

N3 G90 G54 G00 X7.0 Y7.0 S1200 M03

N4 G43 Z25.0 H01 M08

N5 G99 G82 R2.5 Z-3.4 P200 F200.0

N6 M98 P1001

(apelare subprogram)

N7 G80 G00 Z25.0 M09

N8 G28 Z25.0 M05

N9 M01

T02 (burghiu 5mm)

N10 T02

N11 M06

N12 G90 G54 G00 X7.0 Y7.0 S950 M03 T03

N13 G43 Z25.0 H02 M08

N14 G99 G81 R2.5 Z-10.5 F300.0

N15 M98 P1001

(apelare subprogram)

N16 G80 G00 Z25 M09

N17 G28 Z25.0 M05

N18 M01

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

N19 T03 (Tarodare M6X1)

N20 M06

N21 G90 G54 G00 X7.0 Y7.0 S600 M03 T01

N22 G43 Z25.0 H03 M08

N23 G99 G84 R5.0 Z-11.0 F600.0

N24 M98 P1001

(apelare subprogram)

N25 G80 G00 Z25.0 M09

N26 G28 Z25.0 M05

N27 G28 X23.0 Y26.0

N28 M30

%

O1001 (subprogram versiunea 1)

N101 X7.0 Y7.0

(H1)

N102 X39.0

(H2)

N103 Y45.0

(H3)

N104 X7.0

(H4)

N105 X23.0 Y26.0

(H5)

N106 M99 (Sfârșit subprogram)

%

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Subprogram varianta 2

(T01 burghiu de centruire)

```
N1 G21 ;  
N2 G17 G40 G80;  
N3 G90 G54 G00 X7.0 Y7.0 S1200 M03 T02; (H1)  
N4 G43 Z25.0 H01 M08;  
N5 G99 G82 R2.5 Z-3.4 P200 F200.0 L0;  
N6 M98 P1002; (apelare subprogram)  
N7 M01;
```

(T02 burghiu 5 mm)

```
N8 T02;  
N9 M06 ;  
N10 G90 G54 G00 X7.0 Y7.0 S950 M03 T03; (H1)  
N11 G43 Z25.0 H02 M08;  
N12 G99 G81 R2.5 Z-10.5 F300.0 L0;  
N13 M98 P1002; (apelare subprogram)  
N14 M01;
```

Cap.1.  
Introducere

N15 T03

N16 M06

Cap.2.  
Arhitectura  
unui program

N17 G90 G54 G00 X7.0 Y7.0 S600 M03 T01 (H1)

N18 G43 Z25.0 H03 M08

N19 G99 G84 R5.0 Z-11.0 F600.0

Cap.3.  
Limbaaj ISO

N20 **M98 P1002** (apelare subprogram)

N21 G28 X23.0 Y26.0

Cap.4.  
Programarea  
deplasărilor

N22 M30

%

Cap.5.  
Informații  
tehnologice

O1002 (**subprogram varianta 2**)

N101 X7.0 Y7.0 (H1)

N102 X39.0 (H2)

N103 Y45.0 (H3)

N104 X7.0 (H4)

N105 X23.0 Y26.0 (H5)

N106 G80 G00 Z25.0 M09 (Anulare ciclu fix)

N107 G28 Z25.0 M05 (Revenire în HP pe axa Z)

N108 M99 (Sfârșit subprogram)

%

Cap.7.  
Subprograme



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

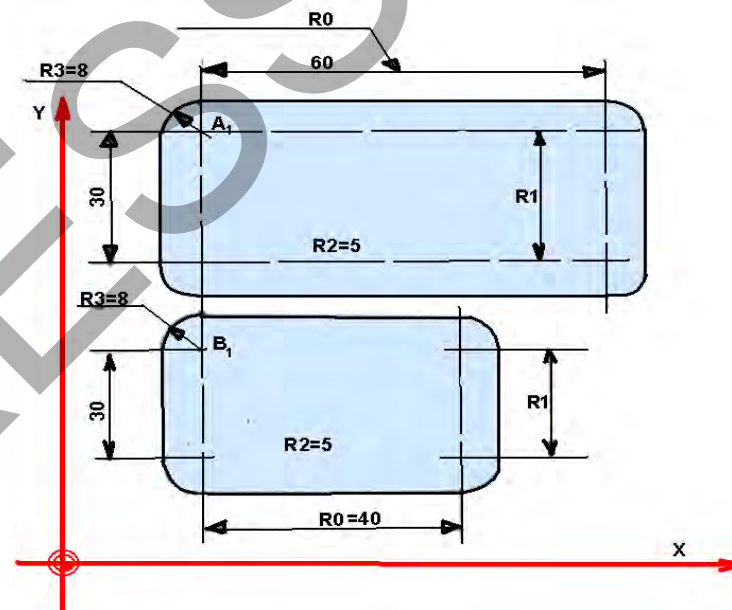
Cap.7.  
Subprograme

# Exemple de subprograme

## Exemplul 1

```
L46
N5  G01 G64  G91 Z = -R2
N10 X = R0
N15  G02 X=R3 Y=-R3  I0 J=-R3
N20  G01 Y=-R1
N25  G02 X=-R0
N30  G01 X=-R0
N35  G02 X=-R3 Y=R3  I0 J=R3
N40  G01 Y=R1
N45  G02 X=R3 Y=R3  I=R3 J0
N50  G01 Z=R2
```

```
LF
LF
LF
LF
LF
LF
LF
LF
LF
LF
LF
LF
```



R2=5 – adâncimea de așchiere

B1 – punctul de început / sfârșit subprogram

% 53 LF

N26 G90 XA1.....YA1 LF (poziția START)

N27 L46 P1 R0=60 R1=30 R2=5 R3=8 LF

N28 G90 XB1.....YB1 LF

N29 L46 P1 R0=40 LF (poziția START)

# Tipuri de subprograme

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## A. TIP BLOC NC

Program principal

%105

N1 G0 Z100

N2 G17 T1 M6

N3 G0 Z2

N4 N\*1 G0 X80 Y50

N5 N\*1 X130 Y80

N6 N\*1 X180 Y50

N7 T0

N8 G0 Z100

N9 T2 M6

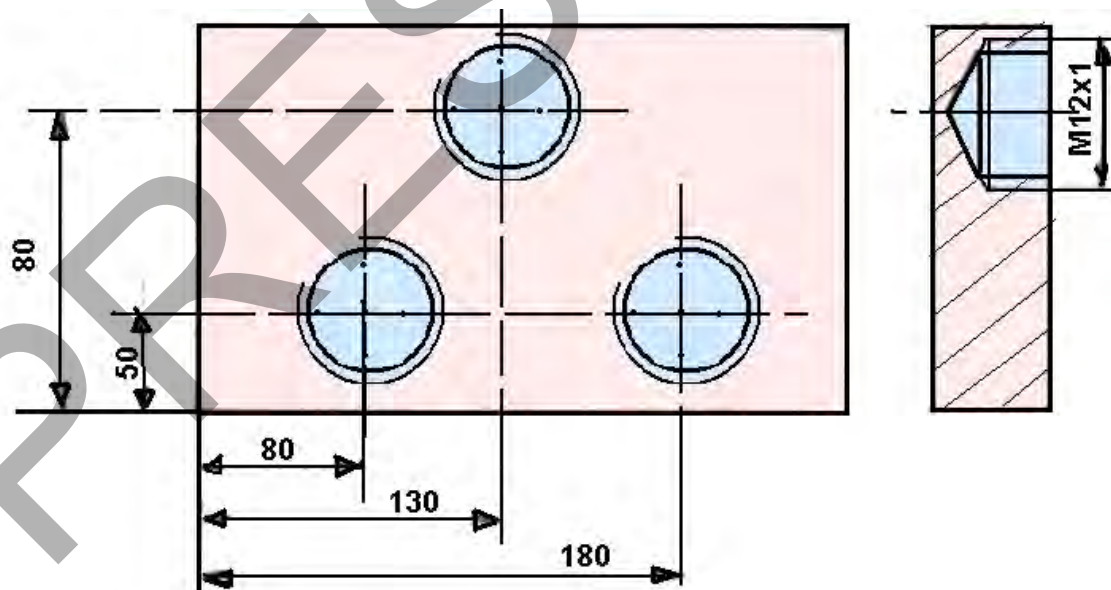
N10 L3 N3 N8 N\*2

N11 T3 M6

N12 G0 Z5

N13 L1 N4 N8 N\*3

N14 M30



N\*1 G81 F200 S3150 Z-4 (op. centruire)

N\*2 G81 F200 S2500 Z-42 (op. burghiere)

N\*3 G84 F200 S200 Z-33 (op. filetare)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

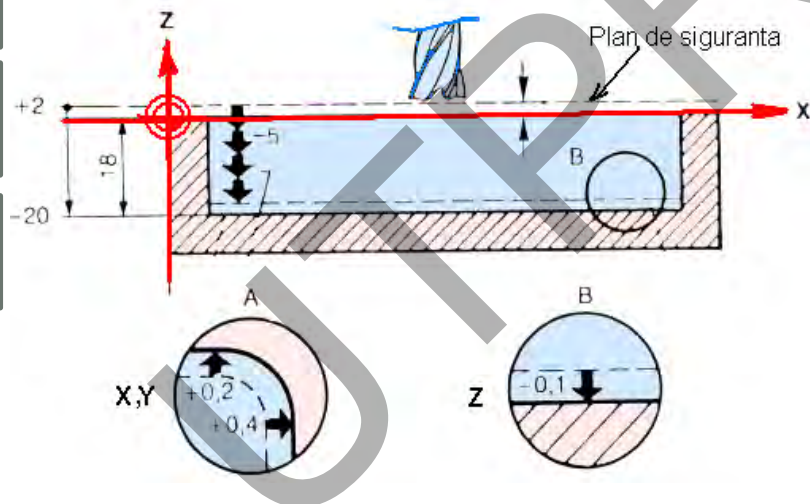
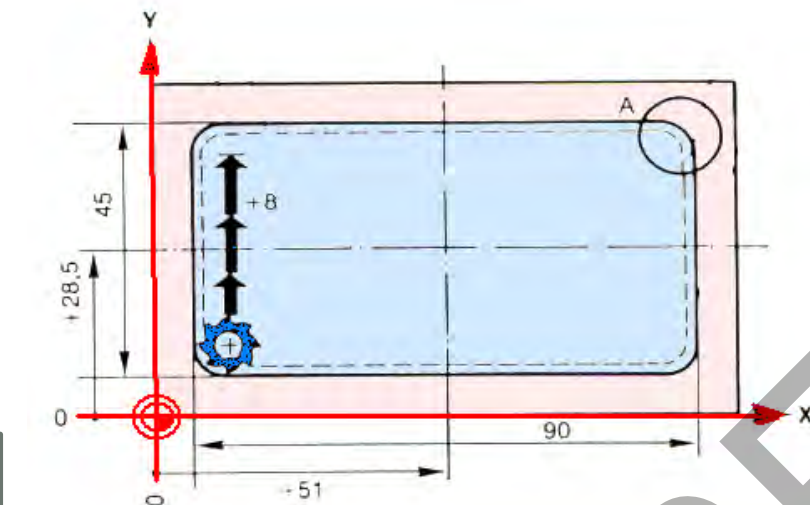
Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme



## Program principal

```
% 200 LF
N1 G17 T1 M3 M6 LF
N5 N*1 G0 X51,0 Y28,5 Z2,0
N10 M30
N*1 G71 F500 S+1000 X90 X0,5 Y45 Y8.0 F100
      Z 20.0 Z-5.0 Z-0.5
N*1 G71 F1__ S__ X1__ X2__ Y1__ Y2__
      F2__ Z1__ Z2__ Z3
```

F1 : avansul de frezare

S+ : codif. Turație

X1 : cota maximă buzunar

X2 : adaos finisare după X,Y

Y1 : cota minimă buzunar

Y2 : lățimea de frezare, după Y

F2 : avans pătrundere

Z1 : adâncimea totală de pătrundere, Z

Z2 : adâncimea de pătrundere, la trecere, pe Z

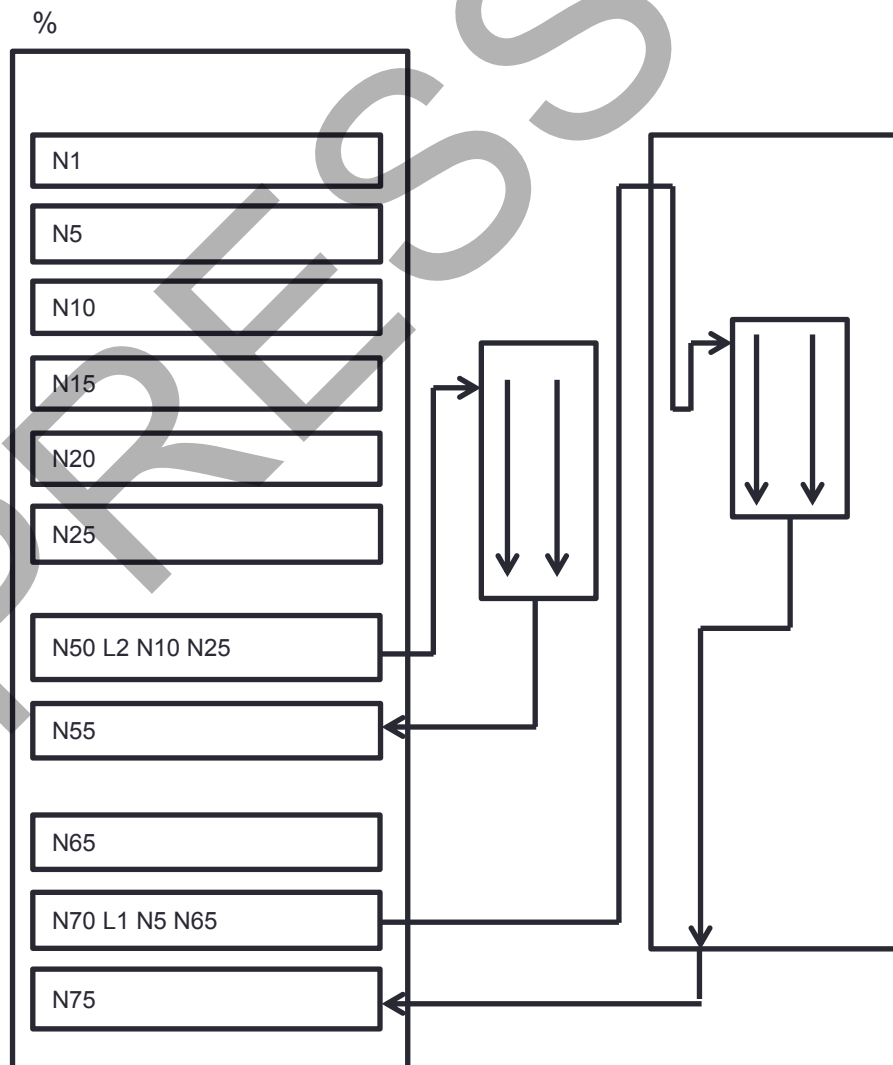
Z3 : adaos de finisare la trecerea buzunarului

# Repetări de blocuri

## Apelare

N50 L2 N... N...

→ nr. de repetări  
(poate fi P)



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Repetări de blocuri

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

%60 LF

N10 G0 Z100 LF

N20 G17T1 M6 LF

N30 G0 X15 Y15 Z2

S2000 M3 LF

N40 G1 Z0 F800

S2000 M3 LF

N50 G91 Z-2 F100 LF

N60 X25 F125 LF

N70 Z-2 F100 LF

N80 X-25 F125 LF

N90 L3 N50 N80 LF Se repetă de 3 ori  
secvențele de prelucrare cuprinse  
între blocurile N50 și N80.

N100 G90 G0 Z2 LF

N110 G0 X60 LF

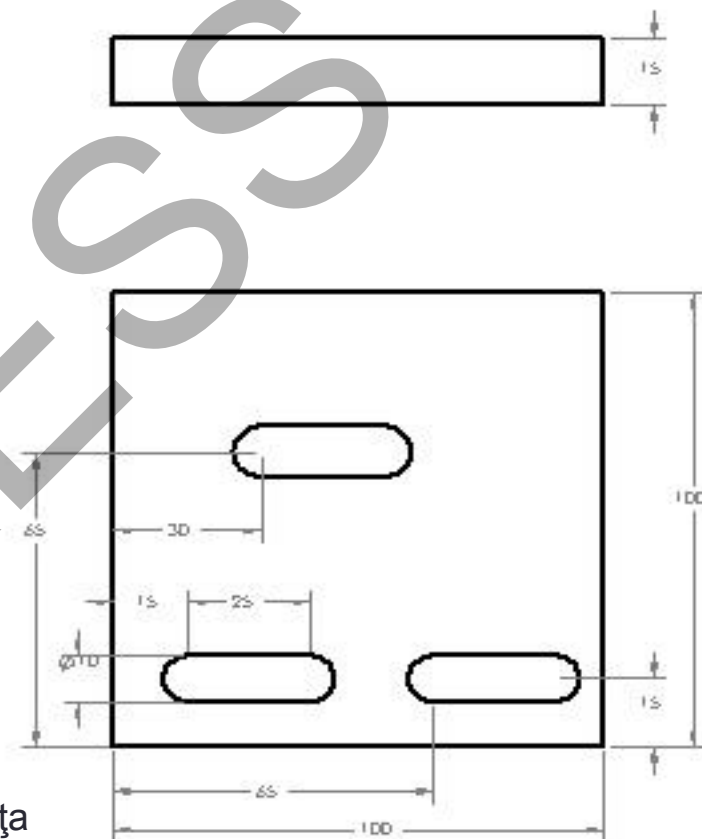
N120 L1 N40 N100 LF Se repetă o dată secvența  
N40-N100.

N130 G0 X30 Y65 LF

N140 L1 N40 N100 LF Se repetă o dată secvența  
N40-N100.

N150 T0 LF

N160 G0 Z100 M30 LF



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Asignare valori

Valoarea parametrilor poate fi asignată sau calculată:

$$P1=17$$

$$P2=3$$

$$P3=(P1+P2)$$

$$P11= (\sin(45))$$

$$P12=\text{SQRT}(80+23)$$

Asignarea valorii parametrilor aritmetici poate fi făcută și în notație exponențială:

$$R1=1.874 \text{ E}x 8 \quad R1=187400000$$

Definirile anterioare nu au un caracter restrictiv, sunt corecte și definirile de tipul

$$P6=Z0$$

$$P7=S1200 F50$$

$$P8= X100.0 Y20.5$$

$$P9=G81$$

Alte notații: P, L Q.

Cap.1.  
Introducere

Valorile numerice ale parametrilor pot fi atribuite unor adrese ca: N, X, Y, Z, I, J, K, L, etc.:

Cap.2.  
Arhitectura  
unui program

N20 G01 XR1/Y2000;

Cap.3.  
Limbaj ISO

Sau

N20 G01 X=R1 Y2000;

Cap.4.  
Programarea  
deplasărilor

Sau

N20 G01 XP1 Y2000;

Cap.5.  
Informații  
tehnologice

Cu ajutorul parametrilor se pot efectua diferite operații:

Cap.6. Cicluri  
fixe

P1=7

P2=3

P3=P1+P2

Cap.7.  
Subprograme

N10 G01 X400.0+R1 Y200-R2 F50+R3;

Majoritatea echipamentelor CNC permit definirea de către utilizator a variabilelor și asignarea valorii acestora.

# Operații aritmetice și logice

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Funcția	Format	Observații
Suma	$\#i = \#j + \#k;$	unghiurile sunt indicate în grade și minute: 90° și 30' se scrie: 90.5
Diferența	$\#i = \#j - \#k;$	
Produs	$\#i = \#j * \#k;$	
Divizare	$\#i = \#j / \#k;$	
Sin	$\#i = \text{SIN}[\#j];$	
Cos	$\#i = \text{COS}[\#j];$	
Tangenta	$\#i = \text{TAN}[\#j];$	
Arcsin	$\#i = \text{ASIN}[\#j];$	
Arccos	$\#i = \text{ACOS}[\#j];$	
Arctangenta	$\#i = \text{ATAN}[\#j];$	
Rădăcina pătrată	$\#i = \text{SQRT}[\#j];$	
Valoare absolută	$\#i = \text{ABS}[\#j];$	
Rotunjire	$\#i = \text{ROUND}[\#j];$	
Logaritm natural	$\#i = \text{LN}[\#j];$	
Funcția exponențială	$\#i = \text{EXP}[\#j];$	
OR	$\#i = \#j \text{ OR } \#k;$	
XOR	$\#i = \#j \text{ XOR } \#k;$	
AND	$\#i = \#j \text{ AND } \#k;$	



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Instrucțiuni macro și NC

În cadrul programării avansate se utilizează atât **instrucțiunile NC** (prezentate anterior) **cât și macrouri** prin care se programează operații aritmetice și logice, diferite instrucțiuni de control (ex. G $\Phi$ T $\Phi$ , D $\Phi$ , END, etc.), blocurile care conțin comenzi pentru apelarea macrourilor.

Celelalte tipuri de blocuri intră în categoria instrucțiuni NC.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Controlul fluxului informațional

Fluxul informațional dintr-un program NC poate fi controlat prin instrucțiunile:

- ❖ G0T0 (salt necondiționat)
- ❖ IF (salt condiționat: IF..., THEN...)
- ❖ WHILE (repetări WHILE)

Exemple:

```
G0T0 1;   G0T0 10;
IF [#1 GT 10] G0T0 2
IF [#1 EQ #2] THEN #3=0    Dacă #1=#2, #3=0
```

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Funcții aritmetice

+ adunare

- scădere

\* multiplicare

/ divizare

Sin ( ) sinus

Cos ( ) cosinus

Tan() tangent

Asin() arcsinus

Acos() arcosinus

Atan() arctangenta

SQRT() rădăcina pătrată

ABS() valoare absolută

## Comparații și operatori logici

### Operatori relaționali

= = egal cu

<> Nu este egal cu

> mai mare decât

< mai mic decât

>= mai mare sau egal

<= mai mic sau egal

### Operatori logici

AND și

OR sau

NOT negație

XOR sau exclusiv

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Salturi și repetări

## ❖ Apelare subprogram

Subprogram

```

% L20 ;
N2001 G01 XR1/;
N2002 G03 XR1/ YR1/ IO JR1/;
N2030 M19;

```

N70 R0/10 R1/20 L2005 ;

N80 G0 G40 M30;

- GO TO < adresa > – salt necondiționat
- IF <expresie> – salt condiționat
- WHILE <expresie> – repetare, până când

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

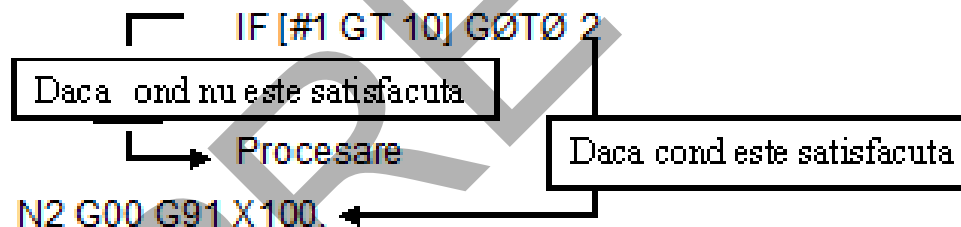
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Salt condiționat IF <expresie>

- Dacă condiția specificată, prin expresie, este satisfăcută se indică locul în care execută saltul



- Dacă condiția specificată este adevărată se execută un macrou predeterminat
- IF [#1 EQ#2]THEN #3=0
- Dacă #1 și #2 sunt identice, valoarea 0 este asignată pentru #3.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

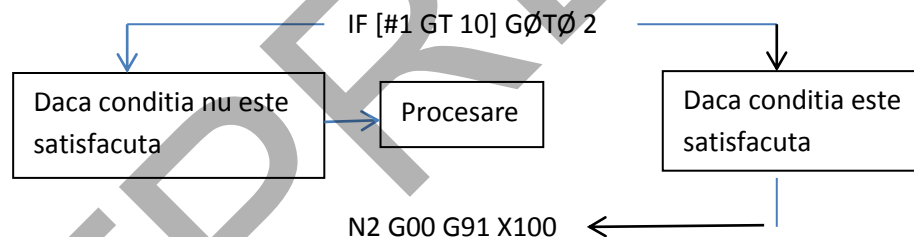
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Salt condiționat IF <expresie>

- **GOTO**

- ❖ Dacă condiția specificată, prin expresie, este satisfăcută se indică locul în care execută saltul
- ❖ Dacă condiția specificată nu este satisfăcută se procesează liniile din program



- **THEN**

IF [#1 EQ#2]THEN #3=0

Dacă #1 și #2 sunt identice,  
valoarea 0 este asignată pentru #3.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Exemplu

09500

#1=0 valoarea inițială a variabilei #1

#2=1 valoarea inițială a variabilei #2

N1 IF[#2 GT10] GOTO 2- salt la N2 când #2 este  
mai mare decât 10

#1=#1+#2 calculul sumei

#2=#2+1 asignarea următoarei valori

GO TO 1 salt la N1

N2 M30 sfârșit program

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

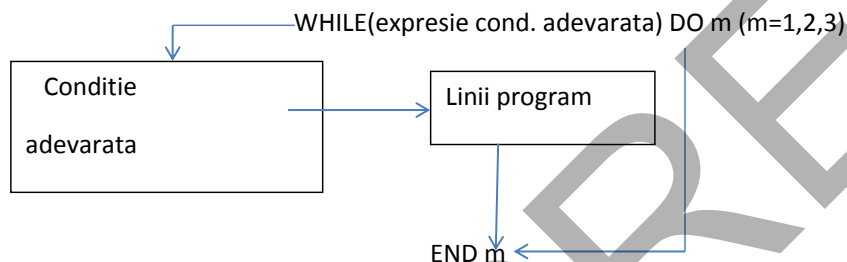
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Repetiție (propoziția WHILE)

Se specifică o anumită condiție după WHILE. Când condiția este îndeplinită se execută programul de la DO la END.

Dacă condiția nu este îndeplinită se realizează un salt în program după END.



### Exemplu

```
O 001;
#1=0
#2=1
WHILE [#2 LE10] DO 1;
#1=#1+#2;
#2=#2+1;
END1;
M30
```



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Operatori utilizați în salturi condiționate

EQ	egal cu (=)
NE	nu este egal (≠)
GT	mai mare decât (>)
LT	mai mic decât (<)
LE	mai mic decât sau egal (≤)

## Exemplu de program

O 9500;	
#1=0	valori inițiale pentru #1 și
#2=1	#2
N1 IF (#2 GT 10) GΦTΦ 2	salt la N2 când #2 este mai mare decât 10
#1=#1+#2	calcularea sumei
#2=#2+1	următoarea însumare
GΦTΦ 1	salt la N1
N2 M30	sfârșit program

# Macrouri

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

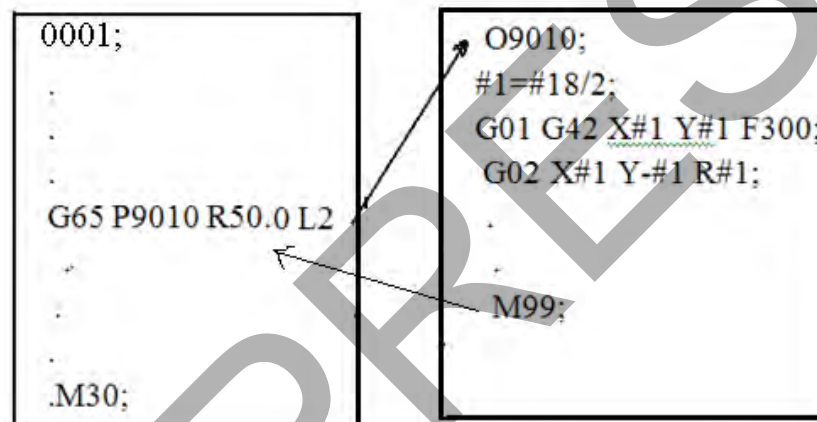
Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Macrourele au întrebunițări similare cu subprogramele. Permit utilizarea variabilelor, funcțiilor logice și aritmetice și a salturilor condiționate.



Blocuri de tip MACRO:

- Blocurile care conțin operații aritmetice și logice
- Blocurile care conțin instrucțiuni de control (ex. GOTO, DO, END, etc.)
- Blocurile care conțin comenzi pentru apelarea macroureilor (G65, G66, G67 sau alte coduri G și M)

Celelalte tipuri de blocuri intră în categoria instrucțiuni NC.

# Macrouri definite de utilizator

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Macrourele permit, față de subprograme, utilizarea de variabile, operații logice și aritmetice, salturi (condiționale, necondiționale) prin care se pot dezvolta ușor programe pentru diferite cicluri fixe.

Apelarea lor se face în programul sursă de prelucrare.

## EXEMPLU

O 001;

...

G65 P9010 R50.0 L2;

...

...

M30

## Macrou

O 9010

#1=#18/2

G01 G42 X#1 Y#1 F300;

G02 X#1 Y-#1 R#1;

...

M99

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## APELARE MACROU

G 65 (apelare nemodală)

G 66 (apelare modală)

G67 (anulare modală)

G65 P<sub>p</sub> L I <specificație, argument>;

P – numărul programului ce se apelează

L – numărul de repetări

argument – date ce se asignează  
macroului

O 001;

...

G65 P9010 L2 A1.0 B2.0

...

...

M30

O 9010;

#3=#1+#2

IF[#3GT360] GOTO 9;

G00 G91 X#3;

N9 M99;

La programarea adresei G65 este apelat macroul definit de adresa P (apelare nemodală)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

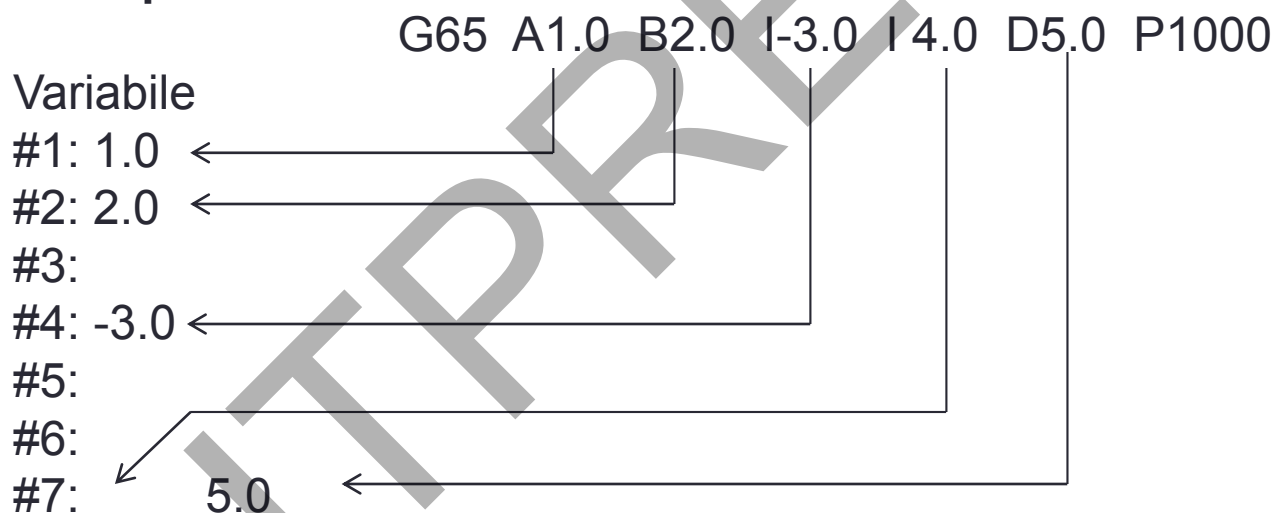
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

G65 trebuie programat înaintea oricărui argument  
Echipamentul CNC identifică argumentele de tip I respectiv II  
În cazul utilizării ambelor tipuri, argumentul (identificat) specificat  
ultimul este prioritar.

### Exemplu



Apelare macrou înlănțuită (nested)

# Specificare argument (asigurare valori)

Tip - I - : Utilizează literele alfabetului cu excepția G,L,N,O,P

Adresa	Variabila	Adresa	Variabila
A	#1	Q	#17
B	#2	R	#18
C	#3	S	#19
D	#7	T	#20
E	#8	U	#21
F	#9	V	#22
H	#11	W	#23
I	#4	X	#24
J	#5	Y	#25
K	#6	Z	#26
M	#13		

Adresele G,L,N,O,P nu pot fi utilizate ca argument;

Adresele care nu trebuie specificate pot fi omise. Variabilele locale corespunzătoare Adreselor omise sunt zero.

Adresele nu trebuie specificate în ordinea alfabetică. Excepție adresele I, J, K.

Exemplu: B\_A\_D\_J\_K\_ - corect

B\_A\_D\_J\_I\_ - incorect

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Tip – II – Utilizează literele A,B,C, o singură dată și literele I,J,K, până la 10 repetări

Adresa	Variabila		Adresa	Variabila
A	#1		K3	#12
B	#2		I4	#13
C	#3		J4	#14
I1	#4		K4	#15
J1	#5		I5	#16
K1	#6		J5	#17
I2	#7		K5	#18
J2	#8		I6	#19
K2	#9		J6	#20
I3	#10		K6	#21
J3	#11		I7	#22

Adresa	Variabila
J7	#23
K7	#24
I8	#25
J8	#26
K8	#27
I9	#28
J9	#29
K9	#30
I10	#31
J10	#32
K10	#33

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Variabile

#1=#2+100;

G01 x#1 F300;

### Tipuri de variabile (CNC, fam. Fanuc)

#1...#33	variabile locale	Pot fi folosite în cadrul macroului pentru a reține rezultatul unor operații. Sunt inițializate la zero la oprirea ECN. La apelarea unui macrou, argumentele sunt asignate prin variabile locale.
#100...#199 #500...#999	variabile comune	Pot fi utilizate de diferite macrouri. La oprirea echipamentului #100...#199 și inițializează la zero, celelalte își păstrează valoarea.
#1000...	variabile sistem	Sunt utilizate pentru a scrie/citi date NC, cum ar fi poziția curentă, corecția de sculă.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Apelare nemodală (G65)

## Formatul de operare

G65 P9100 XxYyZzRrFfliAaBbHh

X- coordonata X a centrului cercului

Y- coordonata Y a centrului cercului

Z-adâncimea găurii

R-coordonatele planului de referință

F-viteza de avans

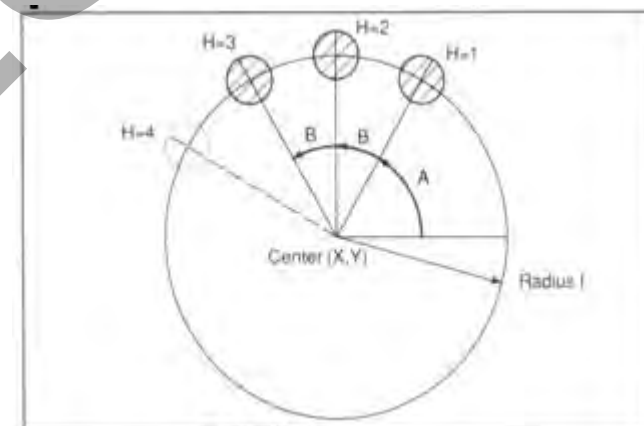
I-raza cercului

A-unghiul de început pentru găurire

B-incrementul unghiular

H-numărul de găuri

P-nume (P=9000-9896)



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

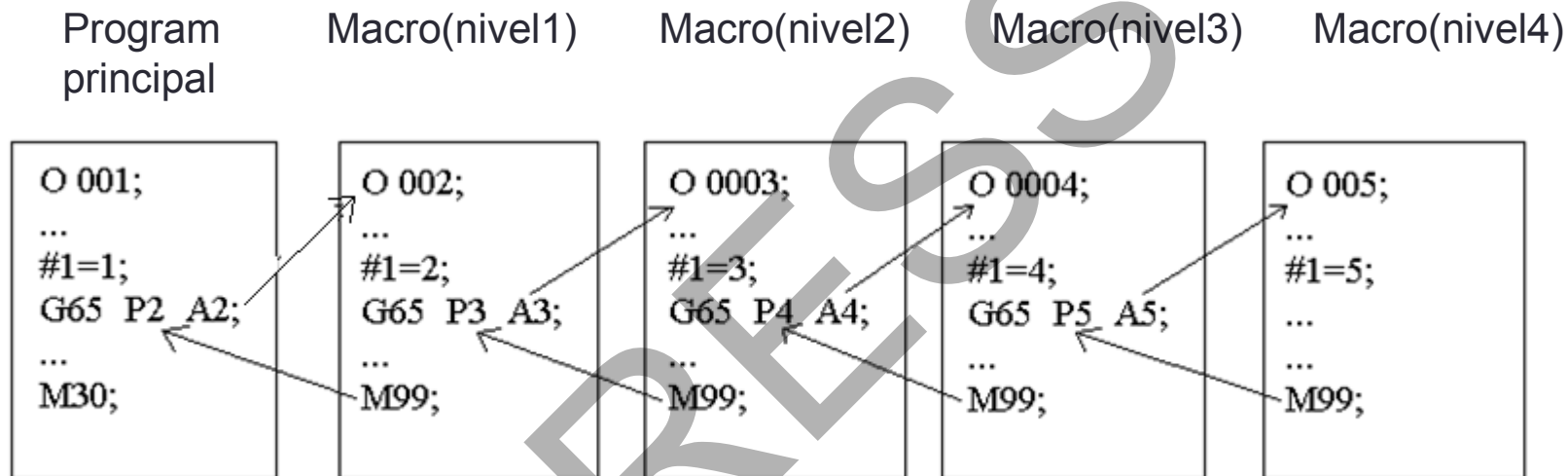
Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

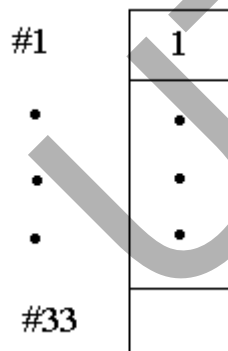
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

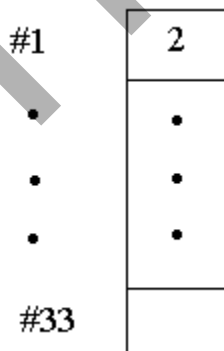
## Apelare macrou înlănțuită (nested)



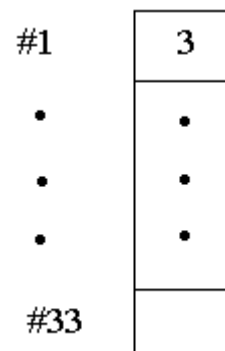
## Variabile locale (nivel 0)



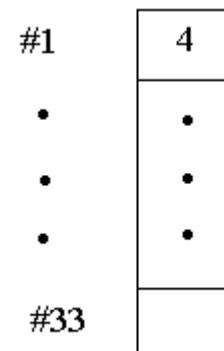
## (nivel 1)



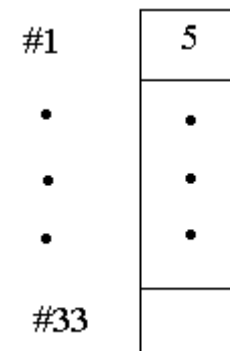
## (nivel 2)



## (nivel 3)



## (nivel 4)



## Program

0002

G90 G92 X0.0 Y0.0 Z100.0;

G65 P9100X100.0Y50.0 R30.0 Z-50.0

F500 I100.0 A0

B45.0 H5;

M30;

### Macrou

O9100;

#3=#4003;..... memorează G grupa 3 (G90, G91)

G81 Z#26 R#18 F#9 K0; Ciclu de găurire

IF [#3EQ90]GOTO1;Salt la N1 pentru G90

#24=#5001+#24; Calculul coordonatei X a centrului;

#25=#5002+#25; Calculul coordonatei Y a centrului;

N1WHILE[#11GT0]DO1; Până nr. de găuri rămase devine zero

#5=#24+#4\*COS[#1]; Calculează poziția X pentru găurire;

#6=#25+#4\*SIN[#1]; Poziția Y pentru găurire;

G90 X#5 Y#6; Executarea găurii după poziționare;

#1=#1+#2; Incrementează unghiul;

#11=#11-1; Decrementează numărul găurilor;

END1;

G#3G80; Revenire la codul G inițial;

M99;

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

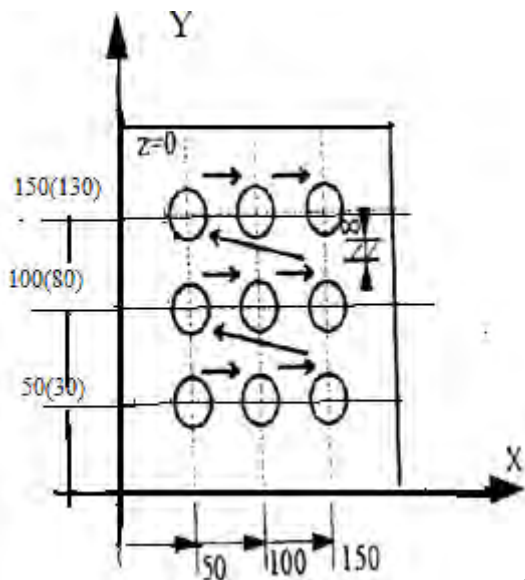
Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

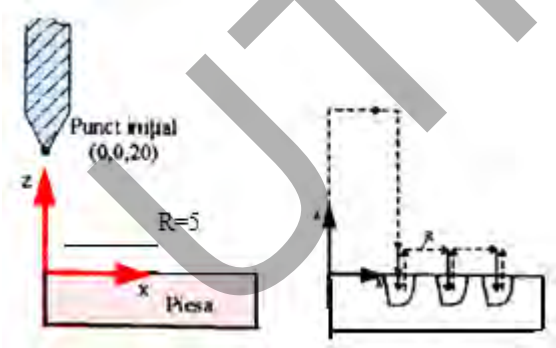
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme



Planul de referință se află  
sub punctul inițial al sculei  
la  $Z=-15$



O 005 program principal nr.5  
N50 G90 X0 Y50 Poziționare pentru primul  
rând de găuri la  $z=20$   
N60 G0 Z20  
N70 G65 P9090 L3 X50 Y0 Z-20 R-15  
T1000 F100 A3 Apelare macrou și  
asignare valori pentru variabile

9090 MACROU

N10 G91

N20 G00 G82 X#24 Y#25 R#26 T#20 F#9  
L#1 Ciclu fix se execută de trei ori ( $A=3$ )  
pentru primul rând de găuri

N30 G00 X[3\*#24]Y#24 Poziționare la  
rândul următor de găuri

N40 G90 M99 Revenire în programul  
principal  
%

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbar ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

## Blocul apelare macrou

N70 G65 P9090 L3 X50 Y0 Z-10 R-15 T1000 F100 A3

- distanța la prima gaură pe axa X (#24)
- distanța de la planul de referință R la suprafața piesei (#26)
- distanța de la poziția inițială la planul de referință R(#18)
- oprire avans la fundul găurii (#20)
- viteza de avans(#9)
- număr de repetări a ciclului fix (#1)

Blocuri din macrou

N20G99G82 X#24 Y#25 Z#26 R#18 P#20 F#9 L#1

N30 G00 X-[3\*#24] Y#24

G99- revenirea sculei în planul de referință R

G82 ciclu fix de găurire

X, Y- distanțe până la găuri

Z- cota găurii, considerând și valoarea de depășire la fund

P- stop la sfârșitul cursei cu avans de lucru

L- număr de repetări

# Apelare modală (G66)

G66 Pp L l <specificație argument, asigurare>; MODALA

P – numărul programului ce va fi apelat;

L – numărul de repetări;

argument: date ce vor fi asigurate variabilelor din macrou;

Anulare caracter model prin G67

Exemplu: Ciclu de găurire, structura de găuri anterioară

## Programul principal

```
O 0001;
...
G66 P9100 L2 A1.0 B2.0;
G00 G90 X100.0;
Y200.0;
X150.0 Y300.0;
G65 ;
...
M30;
```

## Macroul

```
P9100;
...;
G00 Z-#1;
G01 Z-#2 F300;
...
M99;
```

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Apelarea modală are ca efect execuția macroului după fiecare bloc ce conține programată deplasarea până la programarea G67

### Exemplu

Pentru ciclul de burghiere G81

G66 P9110 Xx Yy Zz Rr Ff Ll

X: coordonata X a găurii (#24)

Y: coordonata Y a găurii (#25)

Z: coordonata Z a găurii (#26)

R: coordonata R a planului de referință (#18)

F: viteza de avans (#9)

L: număr de repetări

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

```
O 0001;
G28 G91 X0 Y0 Z0;
G92 X0 Y0 Z50.0;
G00 G90 X100.0 Y50.0;
G66 P9110 Z-20.0 R5.0 F500;
G90 X20.0 Y20.0;
X50.0;
Y50.0;
X70.0 Y80.0 G67;
M30;
```

P9110;

#1=#4001;

#3=#4003;

#4=#4109;

#5=#5003;

G00 G90 Z#18;

G01 Z#26 F#9;

IF [#4010 EQ 98] GΦTΦ 1;

G00 Z#18;

GΦTΦ 2;

N1 G00 Z#5;

N2 G#1 G#3 F#4;

M99;

memorează G00/G01

memorează G90/G91

memorează valoarea F

memorează valoarea Z la începutul burghierii

poziționare în R

deplasare în poziția de început pe Z

revenire în poziția I

revenire în R

poziționare în I

resetare informația modală



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

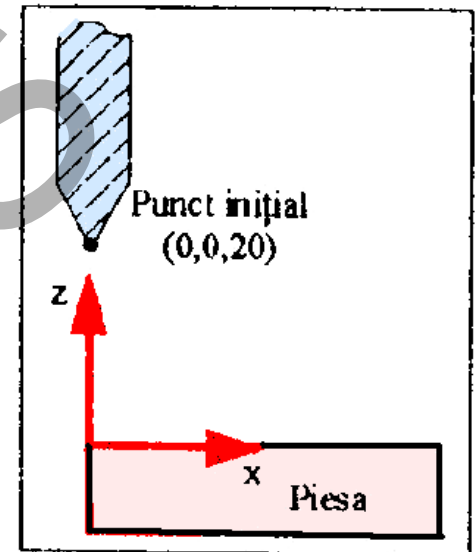
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

# Blocul apelare macrou

N70 G65 P9090 L3 X50 Y0 Z-10 R-15 T1000 F100 A3

- distanța la prima gaură pe axa X (#24)
- distanța la prima gaură pe axa Y (#25)
- distanța de la planul de referință R la suprafața piesei (#26)
- distanța de la poziția inițială la planul de referință R (#18)
- oprire avans la fundul găurii (#20)
- viteză de avans (#9)
- număr de repetări a ciclului fix (#1)



## Blocuri din macrou

N20 G99 G82 X#24 Y#25 Z#26 R#18 P#20 F#9 L#1

N30 G00 X-[3\*#24] Y#24

G99 - revenirea sculei în planul de referință R

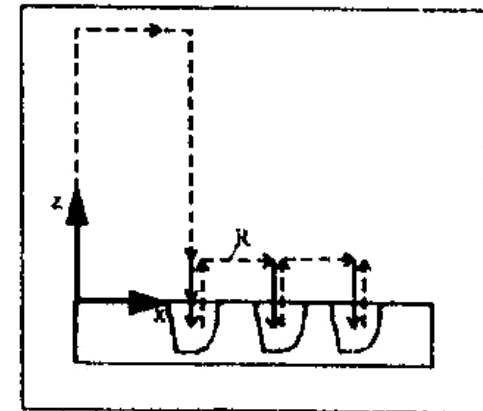
G82 ciclu fix de găurire

X, Y – distanțe până la găuri

Z - cota găurii, considerând și valoarea de depășire la fund

P - stop la sfârșitul cursei cu avans de lucru

L - număr de repetări



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Cap. 8. PROGRAMAREA STRUNGURILOR



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

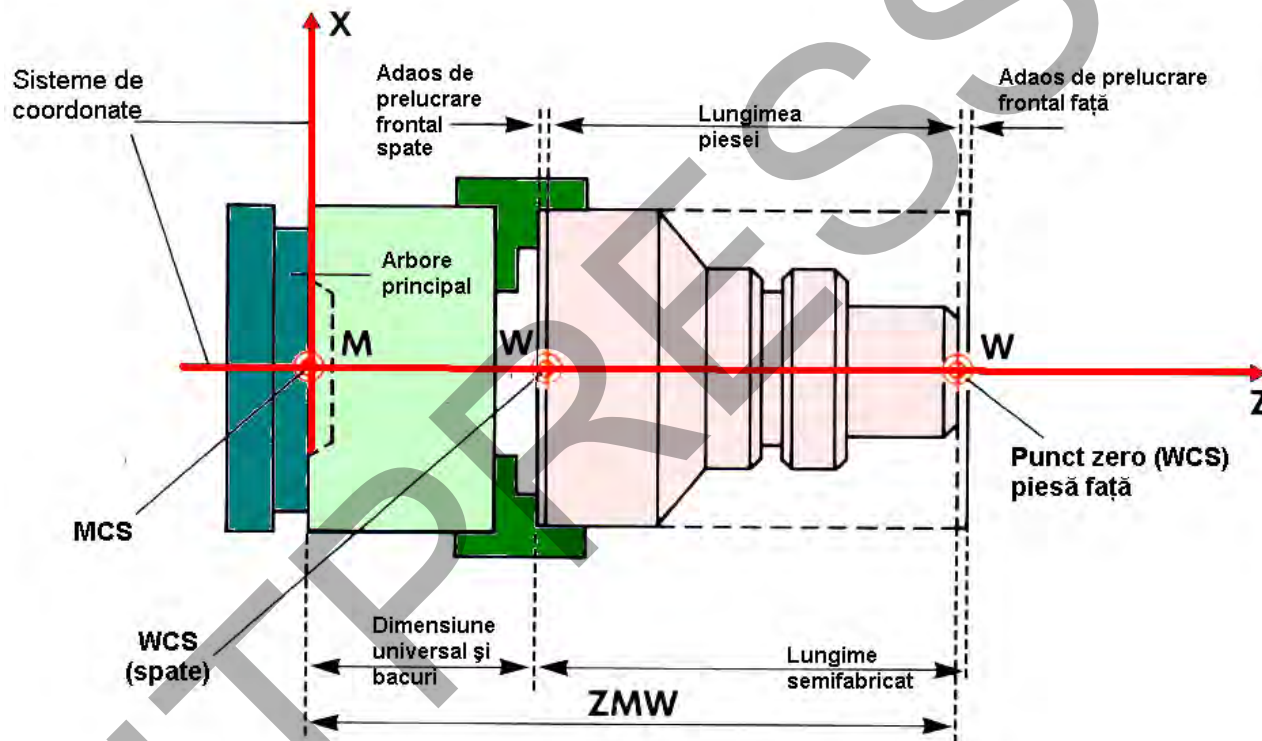
Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

# Puncte de zero și de referință



- Punctul de zero piesă poate fi stabilit în oricare din cele 2 poziții. Se preferă poziția din dreapta.
- Sunt cunoscute dimensiunile universalului și a bacurilor (mărime ZMW – de la originea mașinii la originea piesei).

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

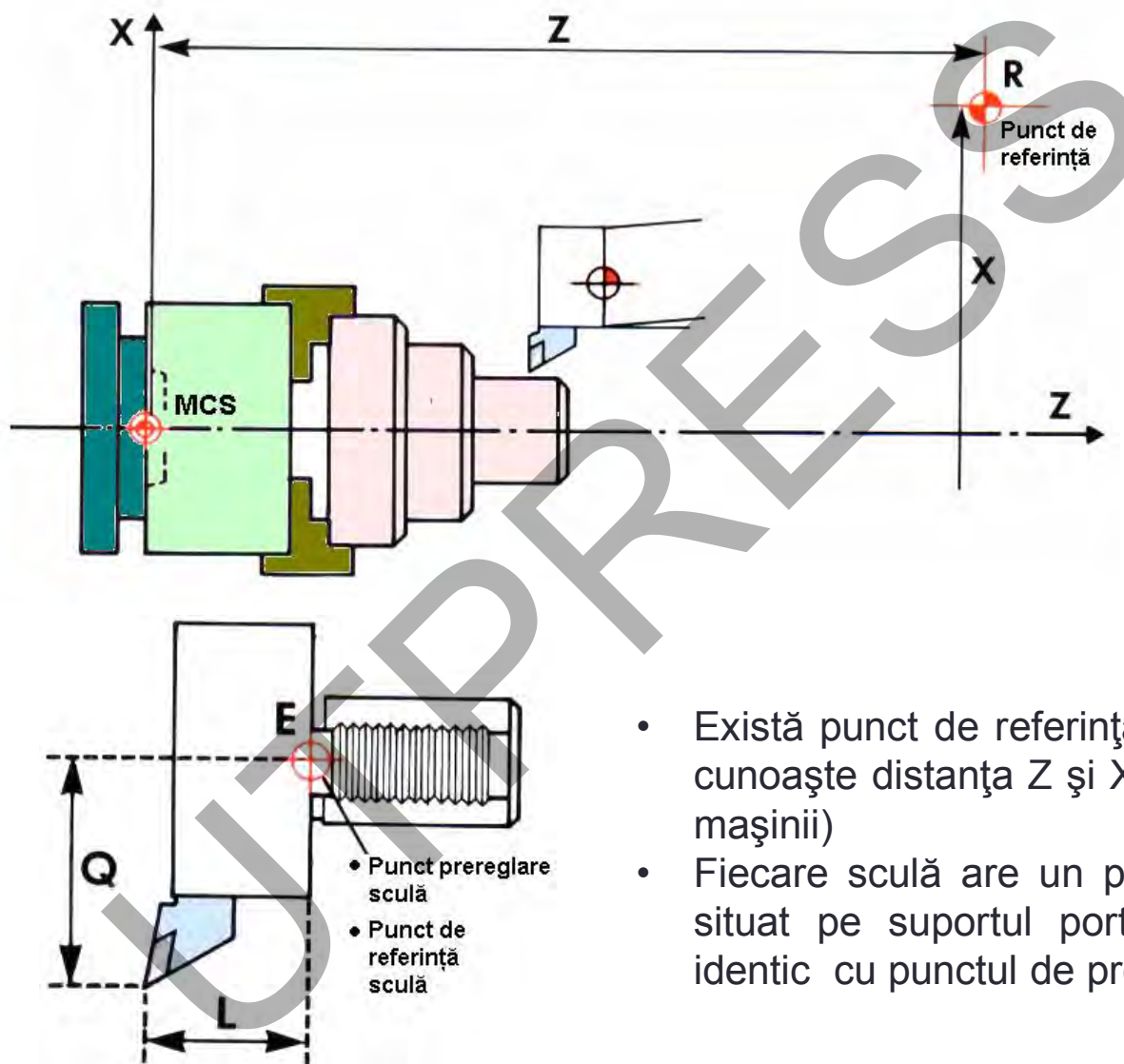
Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor



- Există punct de referință mașină "R" (se cunoaște distanța Z și X față de originea mașinii)
- Fiecare sculă are un punct de referință situat pe suportul port-sculă (T). Este identic cu punctul de prereglare sculă.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

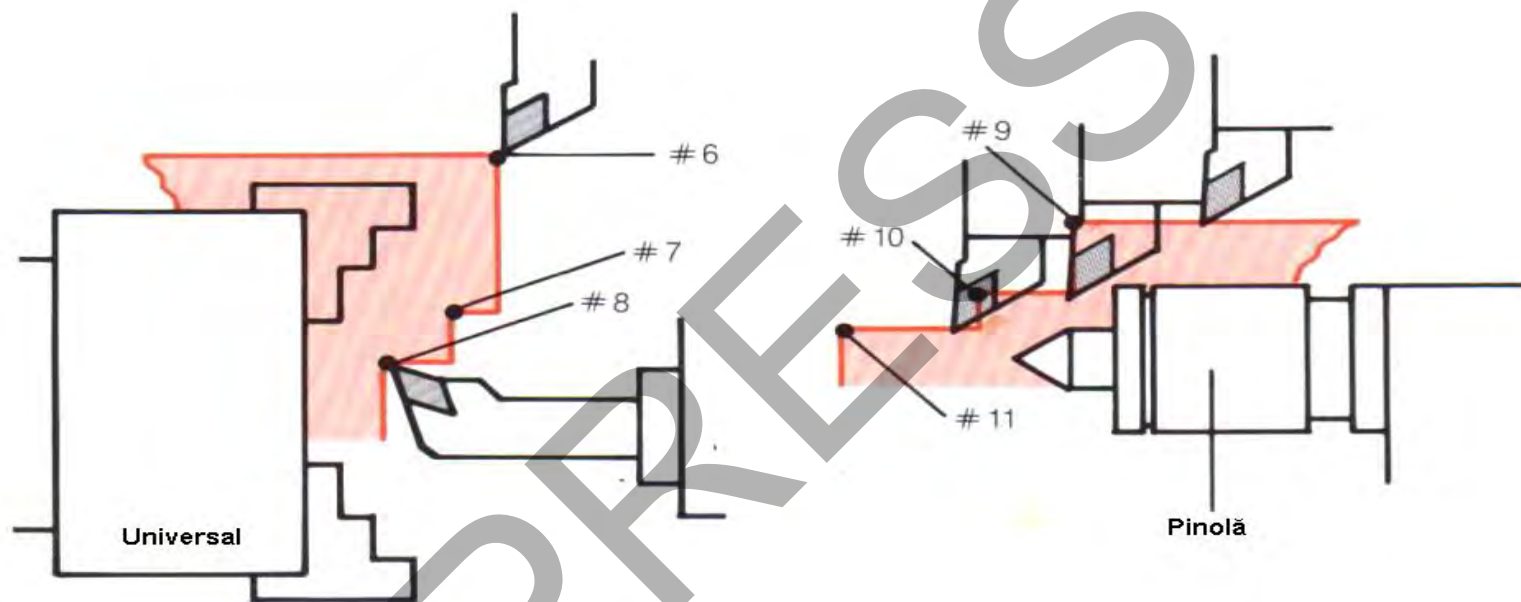
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Zonă interzisă



Pentru evitarea coliziunii între piese, păpușa fixă și sculă, există posibilitatea delimitării unor zone de acces interzis.

Punctele #6 ... #11 sunt definite înaintea începerii prelucrării.

Barierele sunt activate prin funcții "G"

(Ex: G66 activare, G65 dezactivare)

**G65 S ... B ... M ...**

**G66 S ... B ... M ...**



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

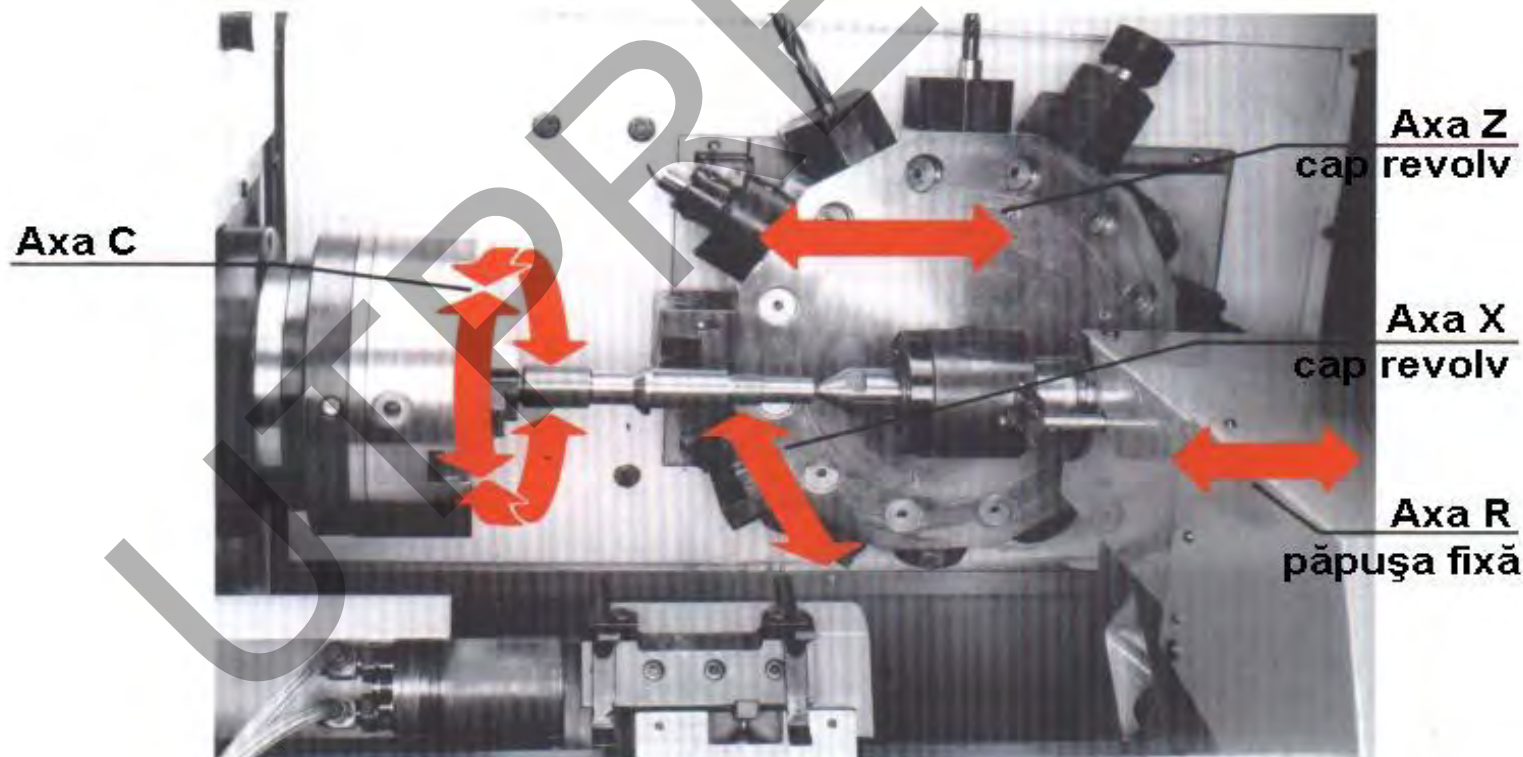
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

# Axe

Centrele de prelucrare prin strunjire au implementate 4 axe:

- axa Z : în lungul AP
- axa X : perpendiculară pe axa Z
- axa C : de rotație în jurul axei Z
- axa R: paralelă cu axa Z (vezi sistemul secundar al axelor de coordonate)



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

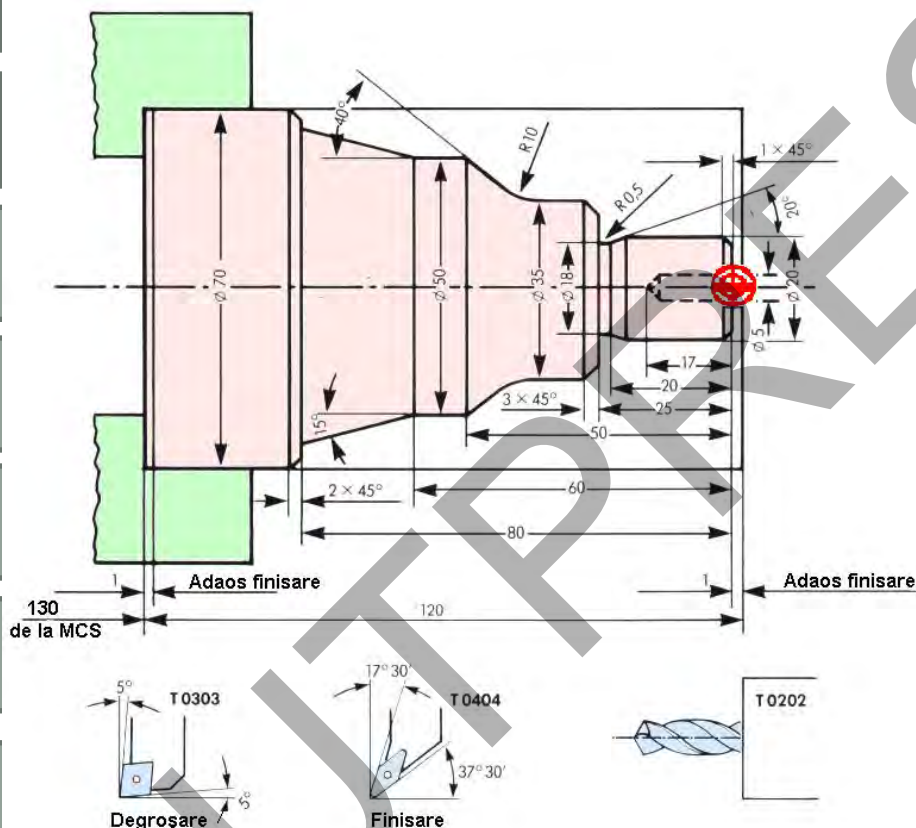
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

# Exemplu de program



## Date inițiale

1. Număr program O 4711
2. Semifabricat  $\text{Ø}70 \times 120$
3. Scule:
  - burghiu, poziția a 2-a din CR
  - cuțit degroșare, poziția a 3-a din CR
  - cuțit finisare, poziția 4 din CR
4. Parametrii regimului de așchiere:
  - burghiere F 0,1 S2000
  - degroșare F 0,4 S200
  - frontal F 0,2 S200
  - finisare F 0,15 S250
5. Zero piesă (G59)
  - universal+bacuri 130 mm
  - lungime semifabricat 120 mm
  - finisare frontală 1 mm
  - deplasare punct zero 249 mm

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

%

O 4711

G59 X0 Z 249 - declarare origine

piesă

(Burghiere T2)

N2 G97 S2000 T0202 M03

G00 X0 Z2 M08

G01 Z-17 F 0.1

G27 M09

(Prelucrare plană și degroșare)

N3 G96 V200 T0303 M04

G00 X74 Z20 M08

G01 X3 F 0.2

G00 X70 Z2

G71 P50 Q60 I 0.5 K 0.1 D 3.5 F

0.4

G26 M09

(Finisare T4)

N4 G96 V250 T0404 M04

N50 G46

G00 X14 Z1 M08

G01 X20 D2 F 0.15

G01 A180

G01 X18 Z-20 A20

G01 Z-25 R 0.5

G01 X35 D3

G01 A180 R10

G01 X50 Z-50 A-40

G01 Z-60

G01 Z-80 A165

G01 X71 D2.5

G01 W-3

N60 G40

G26 M09

M30

%



# Structura programului

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Funcții G.

(sunt prezentate numai cele specifice)

G10 Anulare frezare  
 G11 Frezare  
 G16-G17 Diferite cicluri de frezare  
 G22 Apelare subrutină  
 G24-G27 Deplasare la punct schimbare  
 sculă  
 G33 Filetare  
 G40 Anulare CR  
 G41,G42 Activare CR (cazuri speciale)  
 G46 Activare CR  
 G53 Deplasare rapidă – Punct referință  
 G54-G57 Declarare originea piesei (WPC)  
 G59 Declarare suplimentară WPC  
 G65 Anulare zonă interzisă  
 G66 Activare zonă interzisă  
 G70-G86 Cicluri fixe  
 G94 Avansul în [mm/min]  
 G95 Avansul în [mm/rot]  
 G96 Viteză de așchiere constantă  
 G97 Turația AP  
 G201-G299 Definiere macrouri

Funcții M. - utilizate în derularea  
programului

M00 - Stop Program  
 M99 - Sfârșit subrutină  
 M32 - Activare oglindire  
 M33 - Dezactivare M32  
 M35 - Activare prelucrări conice  
 - arbore principal:  
 M17 - Activare axa C  
 M18 - Dezactivare M17  
 M19 - Frânare axa C  
 M70 - Dezactivare M19  
 -scule rotative:  
 M21, M22 - arborii motori S3, S2/S1  
 M23, M24 - rotație sens orar / antiorar

Funcții B.

B06 - B060000 – B065000  
 rotație sculă între 0-5000 rot/min  
 B07 - B070000 – B070360  
 poziționare unghiulară: 0-360

# Instrucțiuni :

O \_ \_ \_ \_ - nume program (maxim 4 cifre) : ÷ 9999

Tipuri instructiuni:

- Program principal : 1 ... 7999
- Subrutine 8000 ... 8999
- Subrutine Traub 9000 ... 9999
- Macrouri 9800 ... 9899

Exemplu:

O 4711                    Început program  
 G59 X0 Z210  
 G96 V200 T202 M4  
 ...  
 ...  
 M30                      Sfârșit program  
 %

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Program principal cu subrutine încorporate

```

%
O4711
G59X0Z210
G96 V20 T0202 M4
...
...
G22 A8001
...
...
M30
%
```

Diagram illustrating the structure of a main program with subroutines. The main program code is shown on the left, and the subprogram code is shown on the right. A vertical double-headed arrow indicates the range of the subprogram code. A horizontal double-headed arrow indicates the call to the subprogram from the main program.

```

%                               Început subrutină
O8001
G00 X__ Z__
G09 X__
.....
M99
%                               Sfârșit subrutină
```

## Program principal cu macrou

```

%
O4711
...
G201 A__B__C__
...
M30
%
```

Diagram illustrating the structure of a main program with a macro. The main program code is shown on the left, and the macro code is shown on the right. A horizontal double-headed arrow indicates the call to the macro from the main program.

```

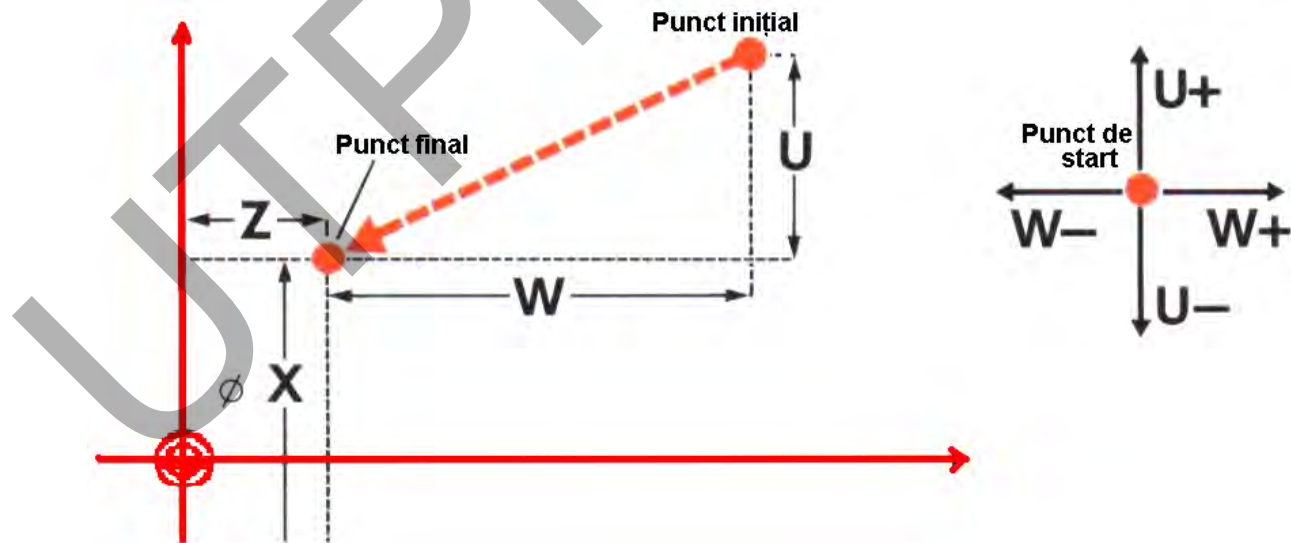
%                               Început macrou
O9801
...
M99                               Sfârșit macrou
```

# Programare deplasări

## Deplasare liniară, avans rapid

$G00$  X/U ... Z/W ... I/A ... K ... F ... S ... B ... T ... M

- funcții M
- Apelare sculă
- Funcții B
- Turatie - Viteză de aşchiere
- Avans, pentru axa R
- Pentru anulare CR
- Coordonatele punctului țintă (final) absolut/incremental



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

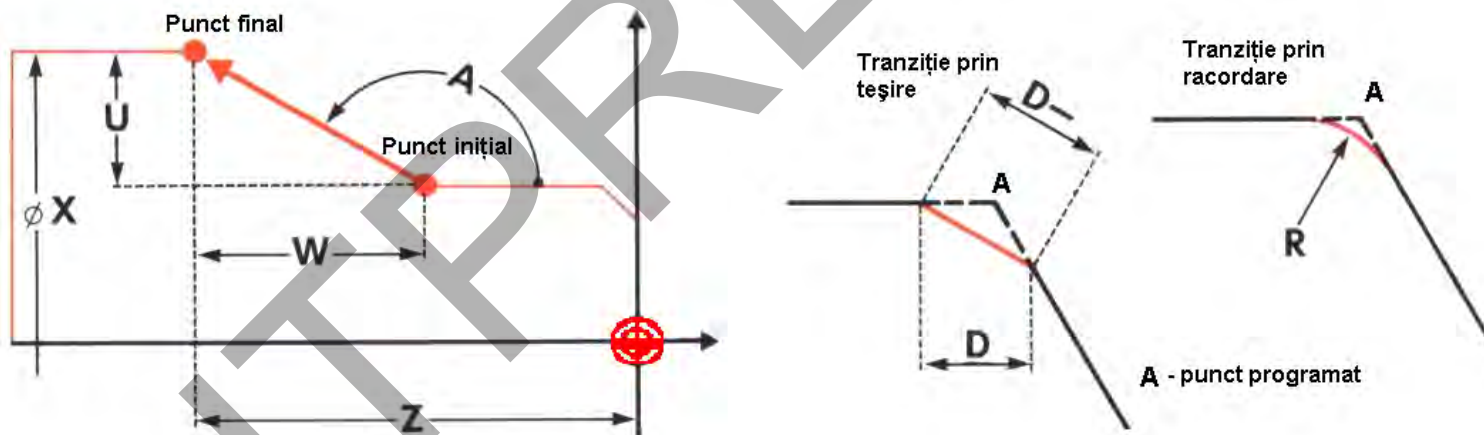
Cap.8. Progra  
marea  
strungurilor

## Deplasare liniară (avans de lucru)

**G01 X/U... Z/W ... A ... D/R ... F ... E ... S ... B ... T ... M ...**



Obs. Celelalte adrese au semnificația indicată anterior



**G09 X/U...Z/W...A...D/R...F...E...S...B...T...M...**

**G09 - are efect de poziționare precisă**



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Deplasare circulară

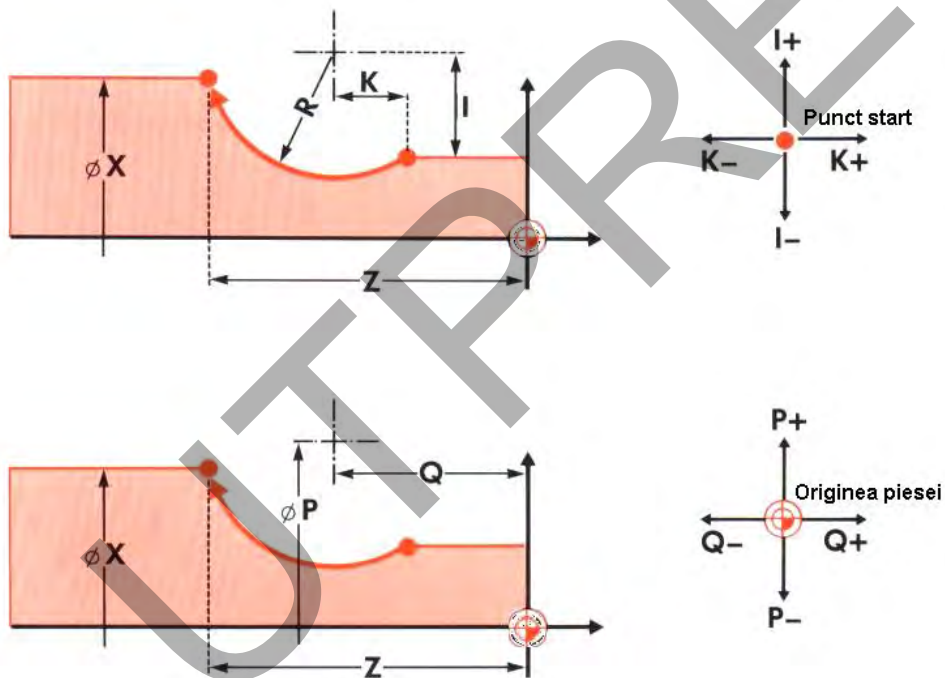
**G02 X/U...Z/W...I...K...D/C...H...F...E...S...B...M...**

**G03**

**P...Q**  
**R...**

Început  
(punct intersecție)

Început (centrul cercului)



Obs. Celelalte adrese au semnificația prezentată anterior

### Programare:

- Prin programarea razei  
G02...X...Z...R  
G03
- Prin programarea  
coordonatelor centrului  
G02...X...Z...P...Q  
G03
- Prin utilizarea  
parametriilor de  
interpolare  
G02...X...Z...I...K...  
G03

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

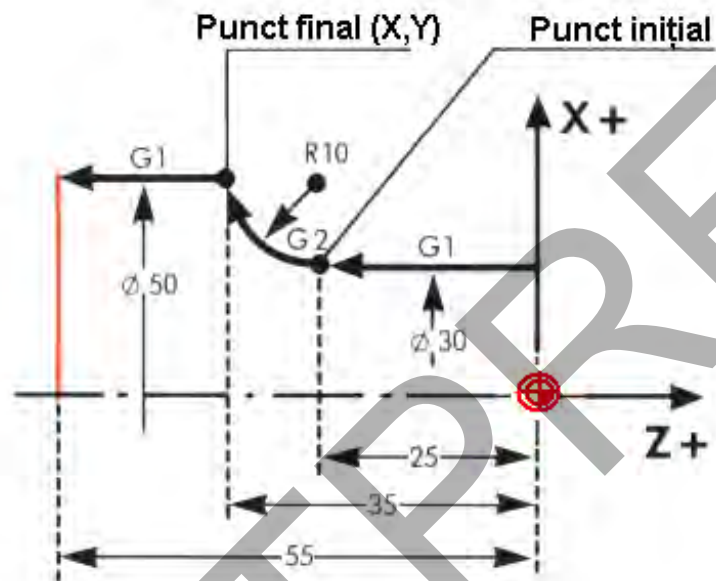
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

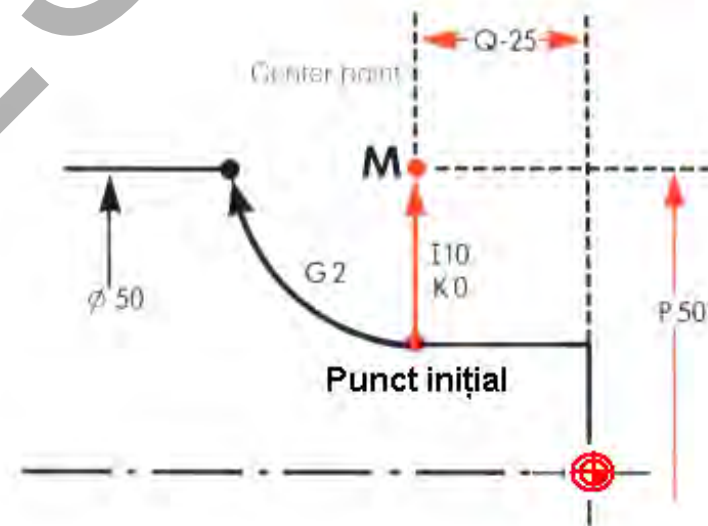
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Exemple:



```
G 0 X 30 Z 1 M 8
G 1 Z - 25 F 0.2
G 2 X 50 Z - 35 R 10
sau G 2 X 50 Z - 35 P 50 Q - 25
sau G 2 X 50 Z - 35 I 10 K 0
G 1 Z - 55
```



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

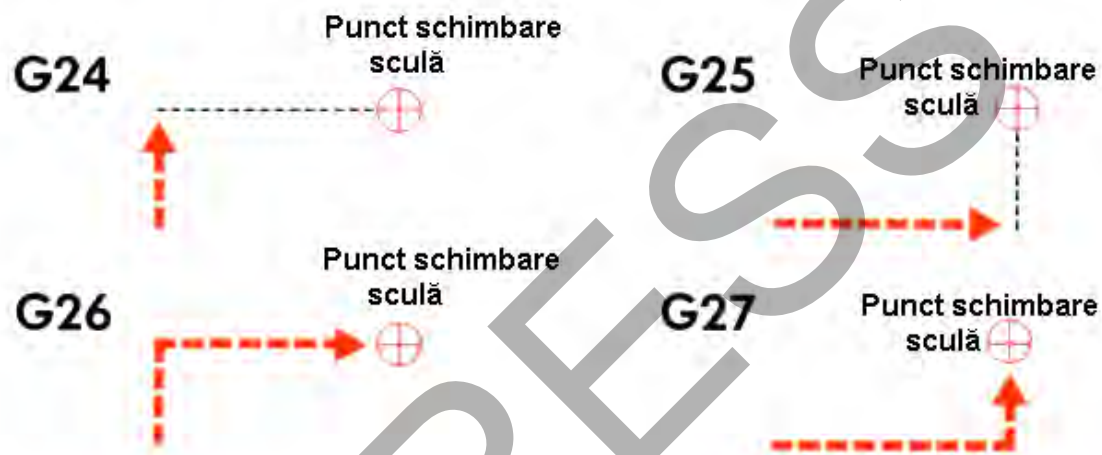
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

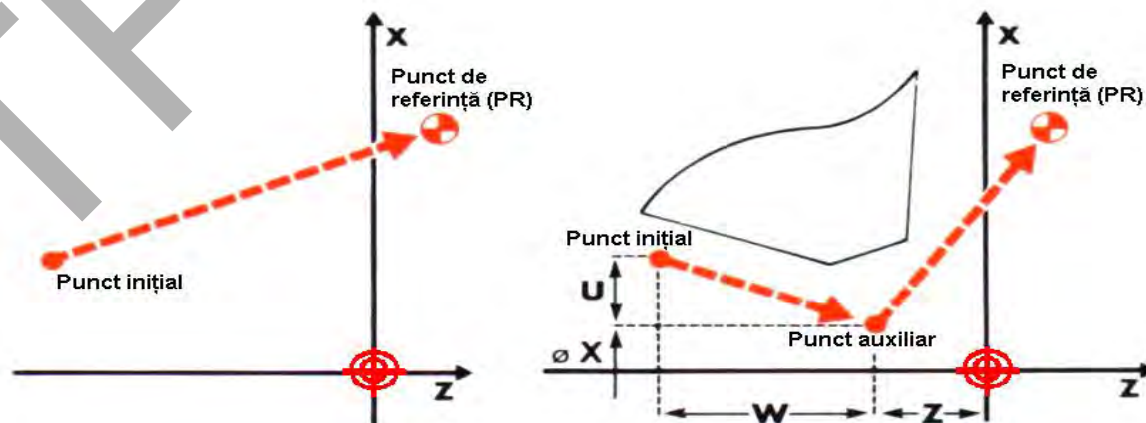
Cap.8. Progra  
marea  
strungurilor

## Deplasare rapidă la punctul de schimbare a sculei



## Deplasare rapidă la punctul de referință

G28 X/U...Z/W...S...M...B...





# Unghiul A

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

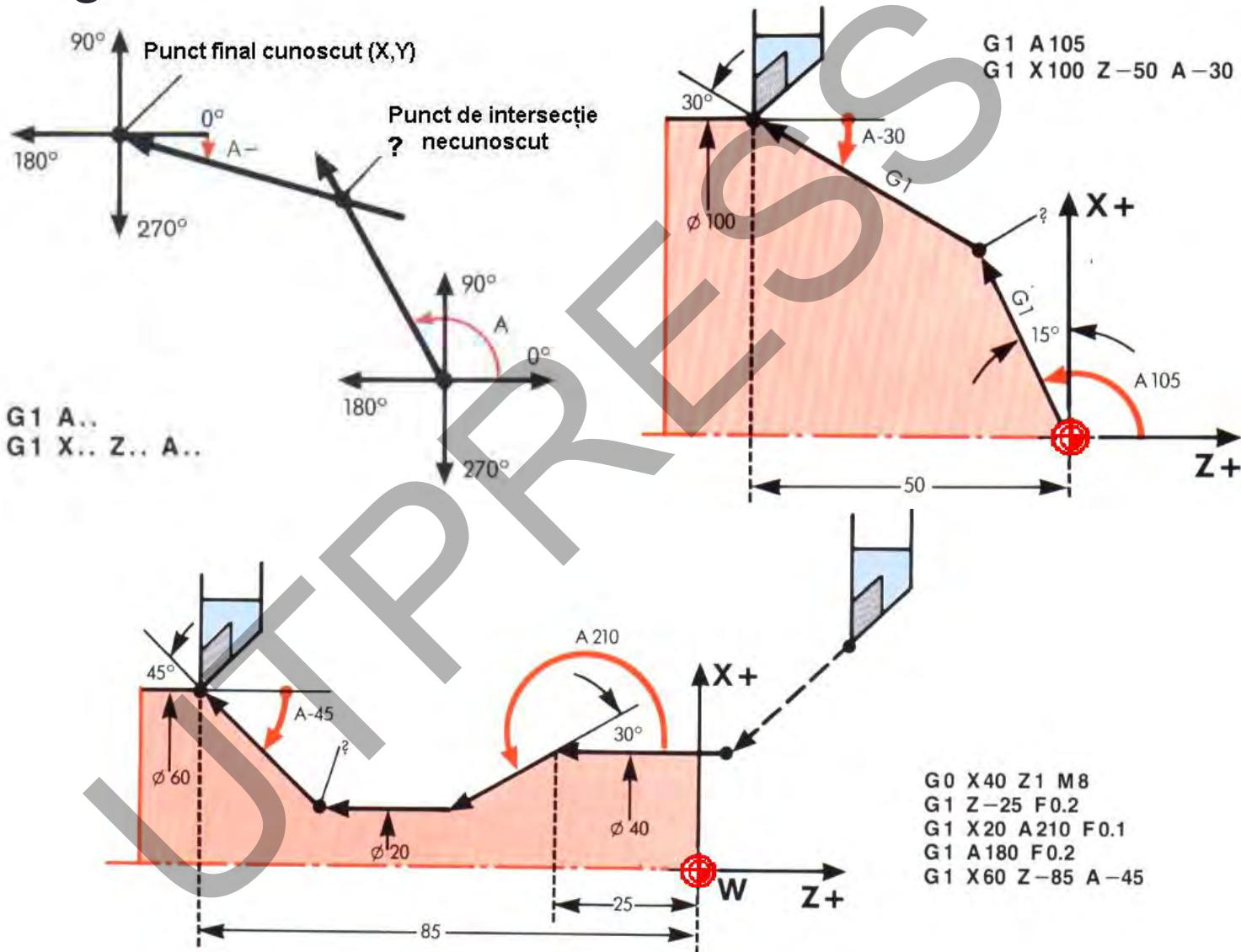
Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

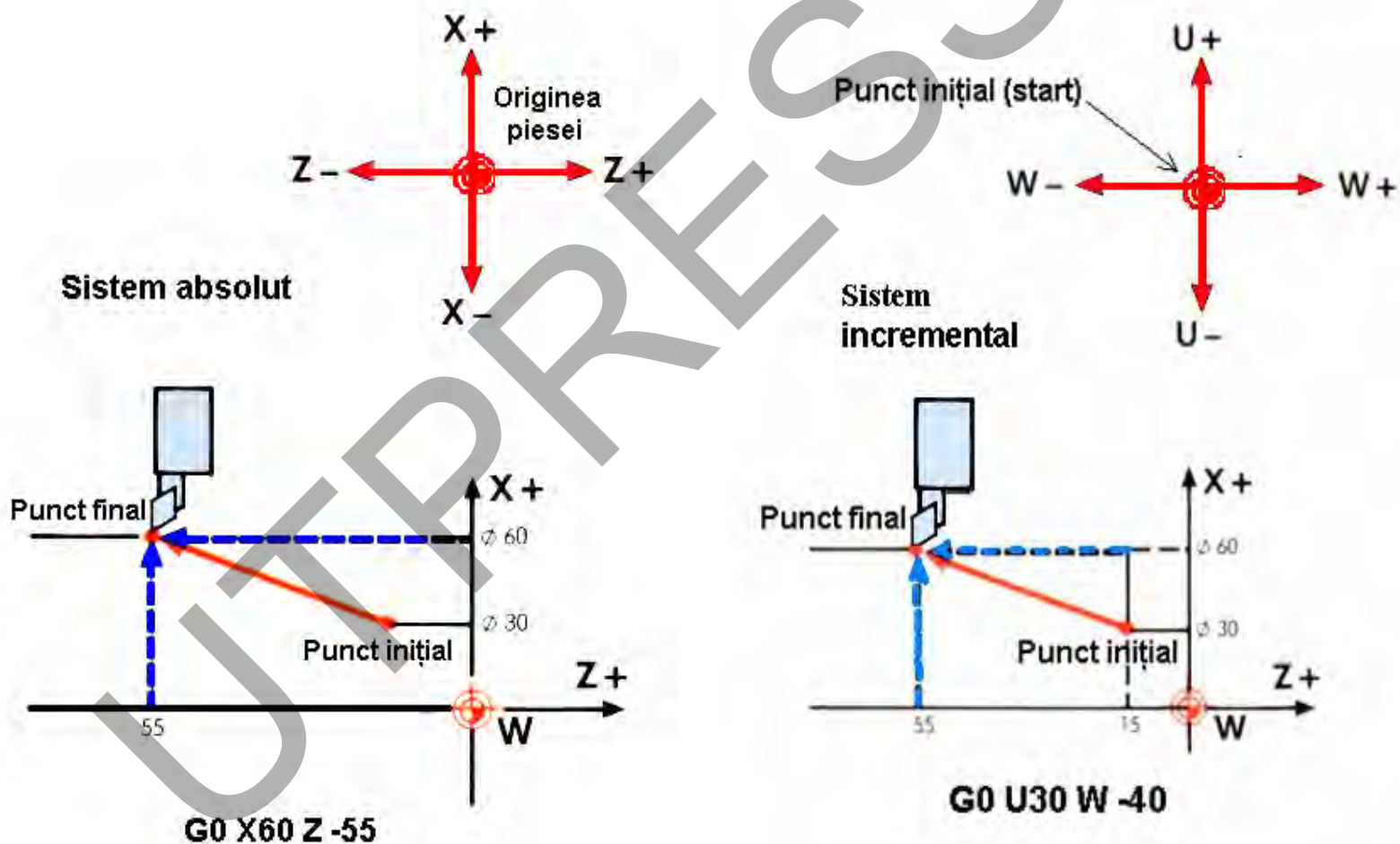
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Programare sistem absolut / incremental



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

# Programarea parametrilor tehnologici

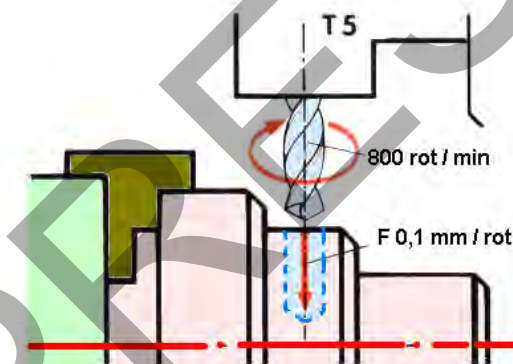
## Programarea avansului

G94 S...B... M...mm/min

Se utilizează :

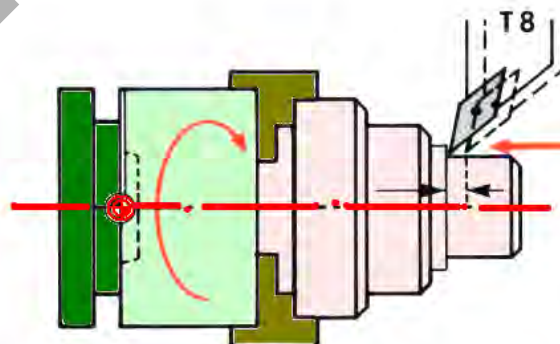
- când AP este oprit
- operații cu axa C
- frezare poligonală
- operații de strunjire

G95...S...B...M...mm/rot



N5 T 0505 M5 M22 B060800  
G 00 Z ...  
G 00 X ... M8  
G 94  
G 01 X ... F80 → avans mm/rot

80 mm/min este echivalent pentru  
n=800 rot/min cu 0,1 mm/rot



N8 G96 V250 T0808 M4  
G 00 X ... Z ... M 08  
G 95  
G 01 Z- ... F0.2

0,2 rot / min

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Programarea turației :

G97 S... / V... X... B... T... M...

S – turația AP

V...X...viteza și poziția AP

Obs. V...X...permite calculul turației:

$$S = \frac{1000 \cdot V}{\pi \cdot X}$$

G96 V... B... T... M...

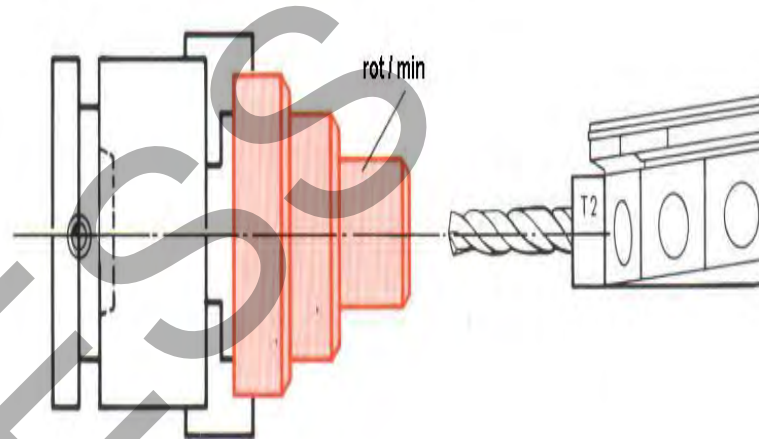
Viteză constantă, strunjiri frontale.

Nu se utilizează pentru găuriri, alezări.

G92 X... Z... S... Q... B... M...

Q – limitare inferioară a vitezei

S – limitare superioară a vitezei



**Pentru suprafețele longitudinale:**

N2 G97 S800 T0202 M3

Sau

N2 G97 V25 X10 T0202 M3

**Pentru suprafețele frontale:**

N6 G96 V25 T0606 M4

G92 S3000

G00 X... Z...

...

G01 Z-... F0.1

...

# Corecția de rază

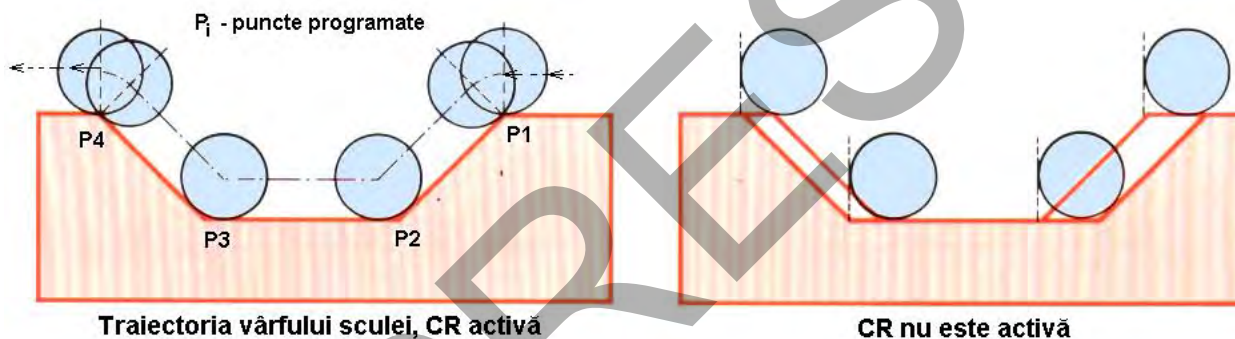
activare

G46 S...B...M...

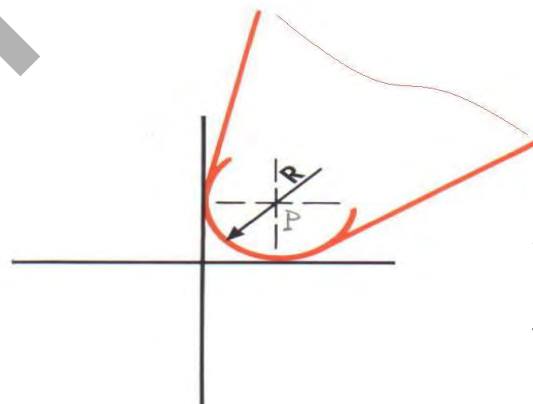
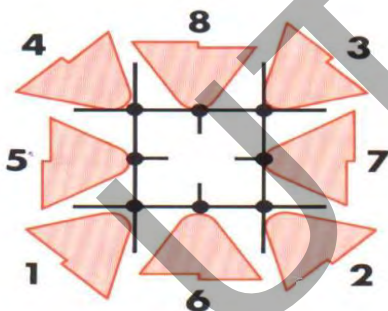
anulare

G40 S...B...M...

**Obs. Funcțiile generale G41, G42 se utilizează mai rar (Traub TX 8D)**



La activare CR cu G46 se introduc în fișierul pentru scule:  
 R – raza la vârful cuțitului  
 poziția centrului-quadrantului, la setarea CNC-ului.



**Notă: Activarea CR cu G41/G42 nu reclamă specificarea quadrantului**

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

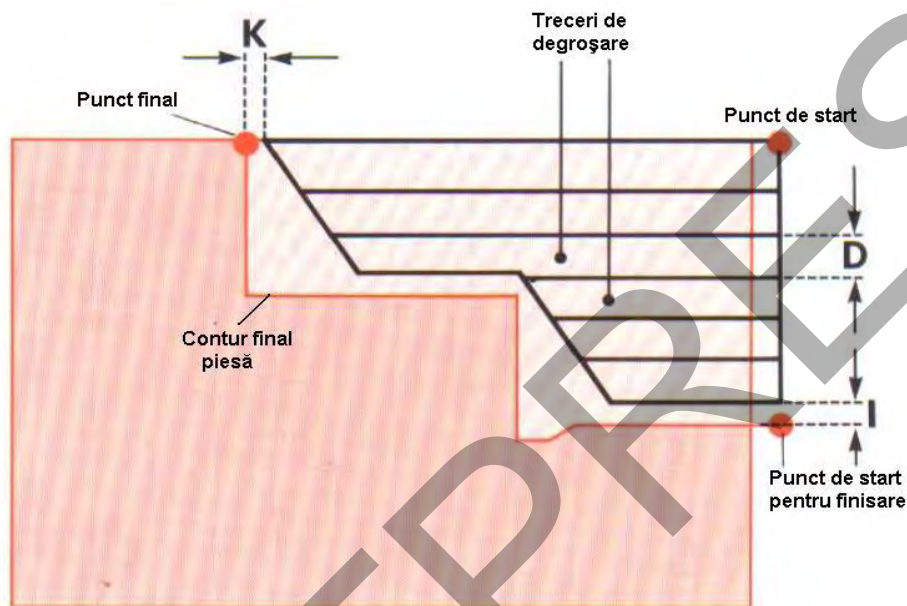
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

# Cicluri fixe

Se prezintă numai ciclurile specifice centrelor pentru strunjire



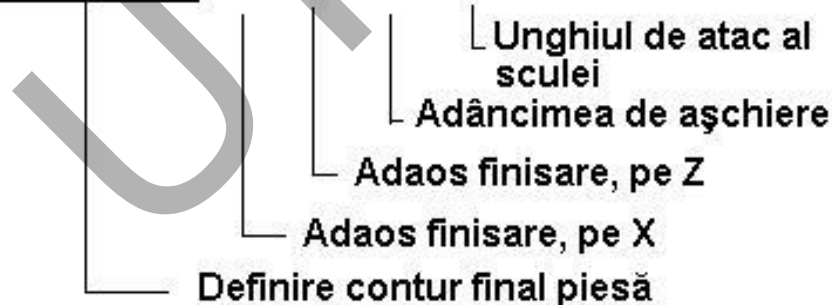
a).

```

...
G96 V180 T505 M04 M08
G0 X100 Z2
G76 P50 Q60 I0,5 K0,05 D4 F0,3
G26
N50 G46 ←
G00 X20 Z1
...
N60 G40 →

```

**G70 A... P... Q... I... K... D... J... F... S...**



b).

```

...
G96 V180 T505 M04 M08
G0 X100 Z2
G70 A520 I0,5 K0,05 D4 F0,3
G26
...
O 520
G46
G0 X20 Z1
...
M99

```

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

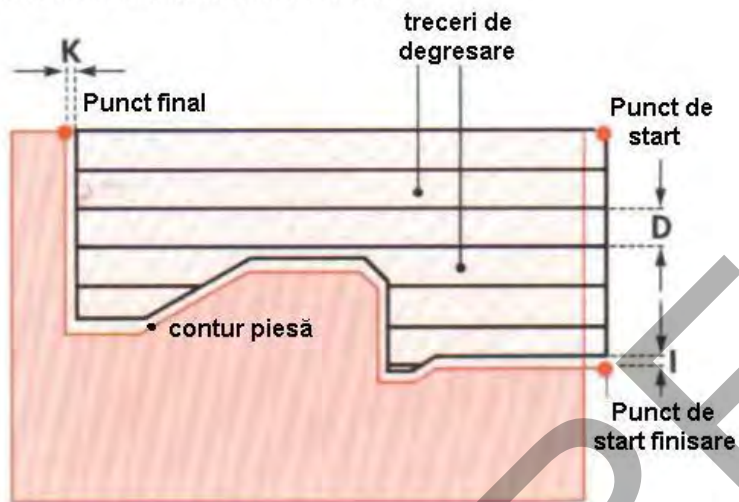
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

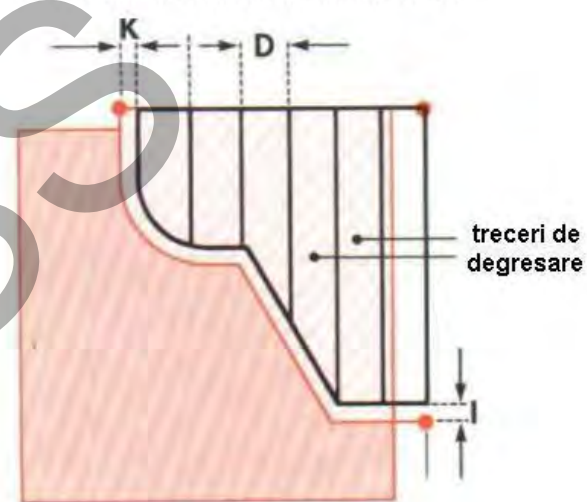
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

G71 A...P...Q...I...K...D...F...E...S...

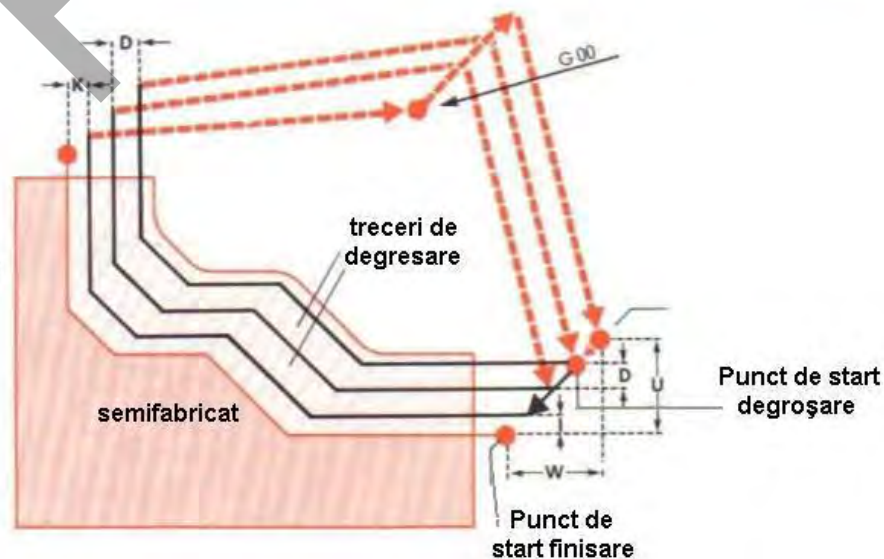


G72 A...P...Q...I...K...D...F...E...S...



G73 A...P...Q...U...W...I/K...D...F...E...S...

Adaosul de pe semifabricat raportat la  
conturul final în direcțiile Z(W) și X(U).



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

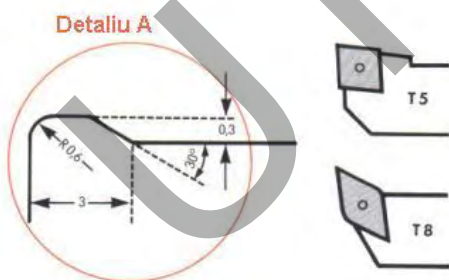
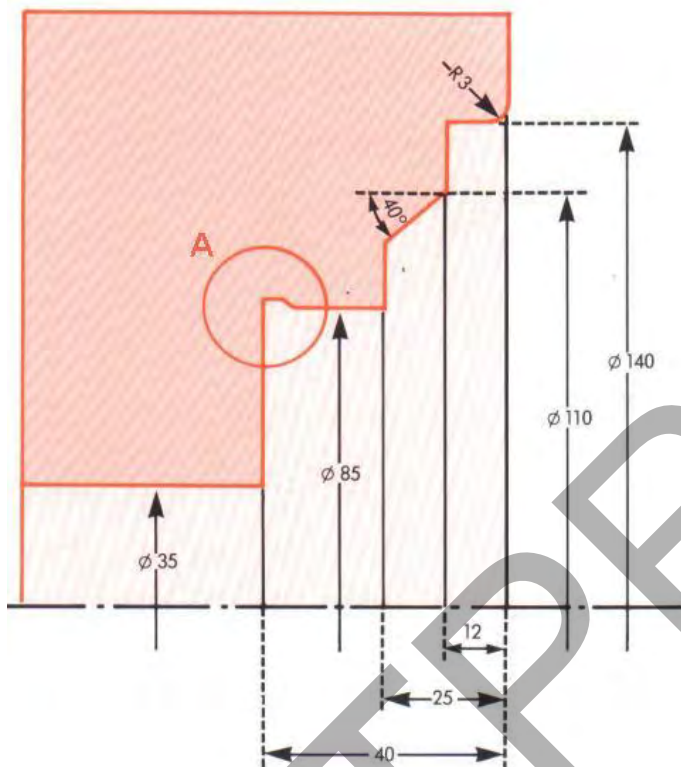
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Exemple: G72



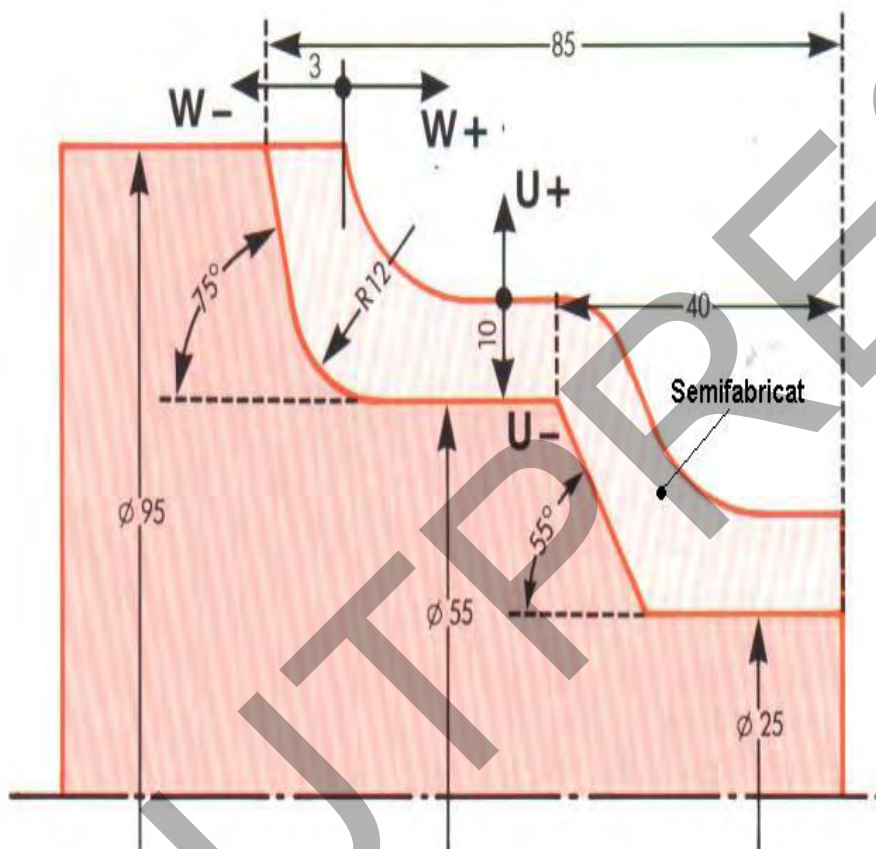
```

...
G96 V140 T505 M04
G0 X30 Z2 M08
G89
G72 A520 I0,5 K0,05 D3,5 F0,3
G27 M09
...
O 520
G46
G0 X33 Z-40
G1 X85
G1 Z-25
G1 A90
G1 X110 Z-12 A220
G1 X140
G1 Z0 R3
G1 X150
G40
M99
...
G96 V180 T808 M4
G46
G0 X150 Z1 M8
G1 Z0 E0,2
G1 X140 R3 E0,1
G1 Z-12
G1 X110
G1 Z-25 A220
G1 X85
G1 Z-37
G1 X85,6 A150 F0,1
G1 Z-40 R0,6
G1 X34
G40
G0 Z10
G26 M09

```



## Exemplu :G73



```

G96 V130 T303 M04
G0 X95 Z1 M08
G73 P50 Q60 I0,5 U-10 W-3 D2 F0,3
G26 M09
...
...
G96 V150 T404 M04
N50 G46
G0 X25 Z1 M08
G1 A180 F0,15
G1 X55 Z-40 A-135
G1 A180 R12
G1 X95 Z-85 A-75
N60 G40
G26 M09
...
M30

```

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

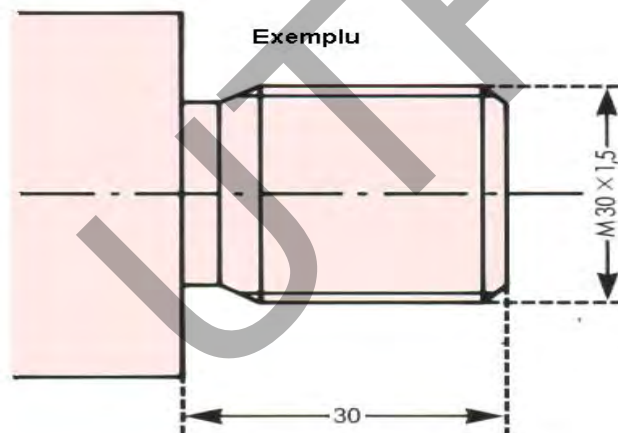
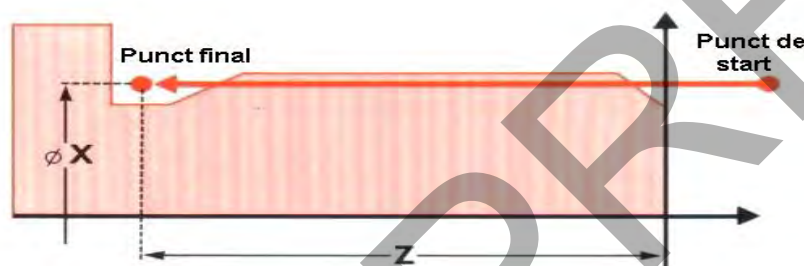
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

# Ciclu de filetare

Se pot realiza filete: - cilindrice, cu pas constant sau variabil  
 - conice, cu pas constant sau variabil  
 - plane  
 - interioare, exterioare

Tehnologie: utilizând cuțite de filetare, prin torodare, filieră;  
 Ciclul cel mai răspândit G33



G33 X/U... Z/W... F/E... B... M...

F/E – pasul filetului:

F : 3 digiti E : 5 digiti

G97 S1000 T404 M03

G0 X29,2 Z4 M08

G33 Z-29,5 F1,5

G0 X35

G0 Z4

G0 X28.8

G33 Z-29,5

G0 X35

G0 Z4

...

G0 X28,16

G33 Z-29.5

G0 X35

G26 M09

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

ISO Extern - Filete metrice

	Pas (mm)	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.75	1.5	1.25	1.0	0.80	0.75	0.50
Adâncimea totală (mm)		3,82	3,52	3,19	2,87	2,53	2,23	1,92	1,60	1,25	1,13	0,93	0,81	0,65	0,52	0,48	0,33
Trecerea 1 (mm)		0,46	0,43	0,41	0,37	0,34	0,34	0,28	0,27	0,24	0,22	0,22	0,21	0,18	0,17	0,16	0,11
2		0,43	0,40	0,39	0,34	0,32	0,31	0,26	0,24	0,22	0,20	0,20	0,17	0,16	0,15	0,14	0,09
3		0,35	0,32	0,32	0,28	0,25	0,25	0,21	0,20	0,18	0,17	0,17	0,14	0,12	0,12	0,11	0,07
4		0,30	0,28	0,27	0,24	0,22	0,21	0,18	0,17	0,16	0,14	0,14	0,11	0,11	0,08	0,07	0,06
5		0,29	0,26	0,24	0,22	0,20	0,18	0,16	0,15	0,14	0,12	0,12	0,10	0,08	---	---	---
6		0,26	0,24	0,24	0,22	0,18	0,18	0,15	0,15	0,12	0,10	0,08	0,08	---	---	---	---
7		0,24	0,21	0,22	0,20	0,17	0,16	0,14	0,12	0,11	0,10	---	---	---	---	---	---
8		0,23	0,20	0,20	0,18	0,15	0,15	0,13	0,11	0,08	0,08	---	---	---	---	---	---
9		0,22	0,19	0,19	0,17	0,14	0,14	0,12	0,11	---	---	---	---	---	---	---	---
10		0,19	0,18	0,18	0,16	0,13	0,12	0,11	0,08	---	---	---	---	---	---	---	---
11		0,18	0,17	0,16	0,14	0,12	0,11	0,10	---	---	---	---	---	---	---	---	---
12		0,16	0,15	0,15	0,13	0,12	0,08	0,08	---	---	---	---	---	---	---	---	---
13		0,15	0,14	0,12	0,12	0,11	---	---	---	---	---	---	---	---	---	---	---
14		0,13	0,13	0,10	0,10	0,08	---	---	---	---	---	---	---	---	---	---	---
15		0,13	0,12	---	---	---	---	---	---	---	---	---	---	---	---	---	---
		0,10	0,10	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Cap.1.  
Introducere**

ISO intern

**Cap.2.  
Arhitectura  
unui program**
**Cap.3.  
Limba ISO**
**Cap.4.  
Programarea  
deplasărilor**
**Cap.5.  
Informații  
tehnologice**
**Cap.6. Cicluri  
fixe**
**Cap.7.  
Subprograme**
**Cap.8. Progra  
marea  
strungurilor**

Pas (mm)	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0	1.75	1.5	1.25	1.0	0.8	0.75	0.50
Adancimea totala (mm)	3.54	3.25	2.96	2.65	2.33	2.05	1.78	1.48	1.17	1.05	0.85	0.75	0.6	0.49	0.46	0.31
Trecerea 1 (mm)	0.46	0.43	0.42	0.37	0.34	0.32	0.28	0.26	0.23	0.22	0.20	0.17	0.17	0.17	0.16	0.10
2	0.43	0.4	0.4	0.34	0.31	0.3	0.26	0.25	0.21	0.2	0.18	0.17	0.15	0.14	0.13	0.08
3	0.35	0.33	0.32	0.28	0.24	0.24	0.21	0.18	0.17	0.15	0.15	0.14	0.11	0.11	0.10	0.07
4	0.3	0.26	0.26	0.23	0.21	0.19	0.16	0.15	0.15	0.13	0.13	0.10	0.09	0.07	0.07	0.06
5	0.26	0.22	0.22	0.21	0.18	0.17	0.14	0.13	0.12	0.10	0.11	0.09	0.08	-	-	-
6	0.22	0.2	0.2	0.19	0.15	0.15	0.13	0.12	0.11	0.09	0.08	0.08	-	-	-	-
7	0.2	0.18	0.17	0.16	0.14	0.14	0.12	0.11	0.10	0.08	-	-	-	-	-	-
8	0.19	0.17	0.16	0.15	0.13	0.13	0.11	0.10	0.08	0.08	-	-	-	-	-	-
9	0.18	0.16	0.16	0.14	0.12	0.12	0.1	0.1	-	-	-	-	-	-	-	-
10	0.16	0.15	0.15	0.13	0.12	0.11	0.10	0.08	-	-	-	-	-	-	-	-
11	0.15	0.14	0.14	0.12	0.11	0.10	0.09	-	-	-	-	-	-	-	-	-
12	0.15	0.14	0.14	0.12	0.10	0.08	0.08	-	-	-	-	-	-	-	-	-
13	0.14	0.13	0.12	0.11	0.10	-	-	-	-	-	-	-	-	-	-	-
14	0.13	0.12	0.1	0.1	0.08	-	-	-	-	-	-	-	-	-	-	-
15	0.12	0.12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	0.1	0.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Ciclu de filetare scule Sandvik



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

# Strungire: Aplicația 1

## Tipuri de scule utilizate la operațiile de strungire:

-Diferite tipuri de plăcuțe

-Oțel rapid

### Plăcuțe:

V- sunt indicate pentru prelucrări de finisare, 2 muchii accesibile puțin rezistente

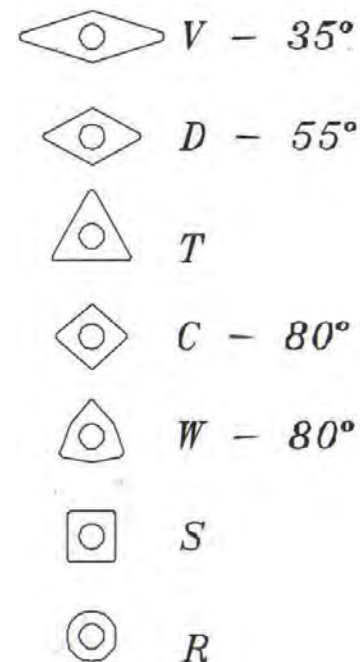
D- sunt indicate pentru profilări (dacă permite unghiul de 55) mai rezistente, 2 muchii accesibile

T- frecvent utilizate, au 3 muchii accesibile

C- frecvent utilizate deoarece dispozitivul de fixare al plăcuței poate fi utilizat pentru strunjiri longitudinale și plane, 2 muchii

W,S- foarte rezistente, utilizate în special pentru degajări, 4 muchii accesibile

R- cea mai rezistentă, puțin utilizată.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Cuțite pentru strunjire, filetare, retezare



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

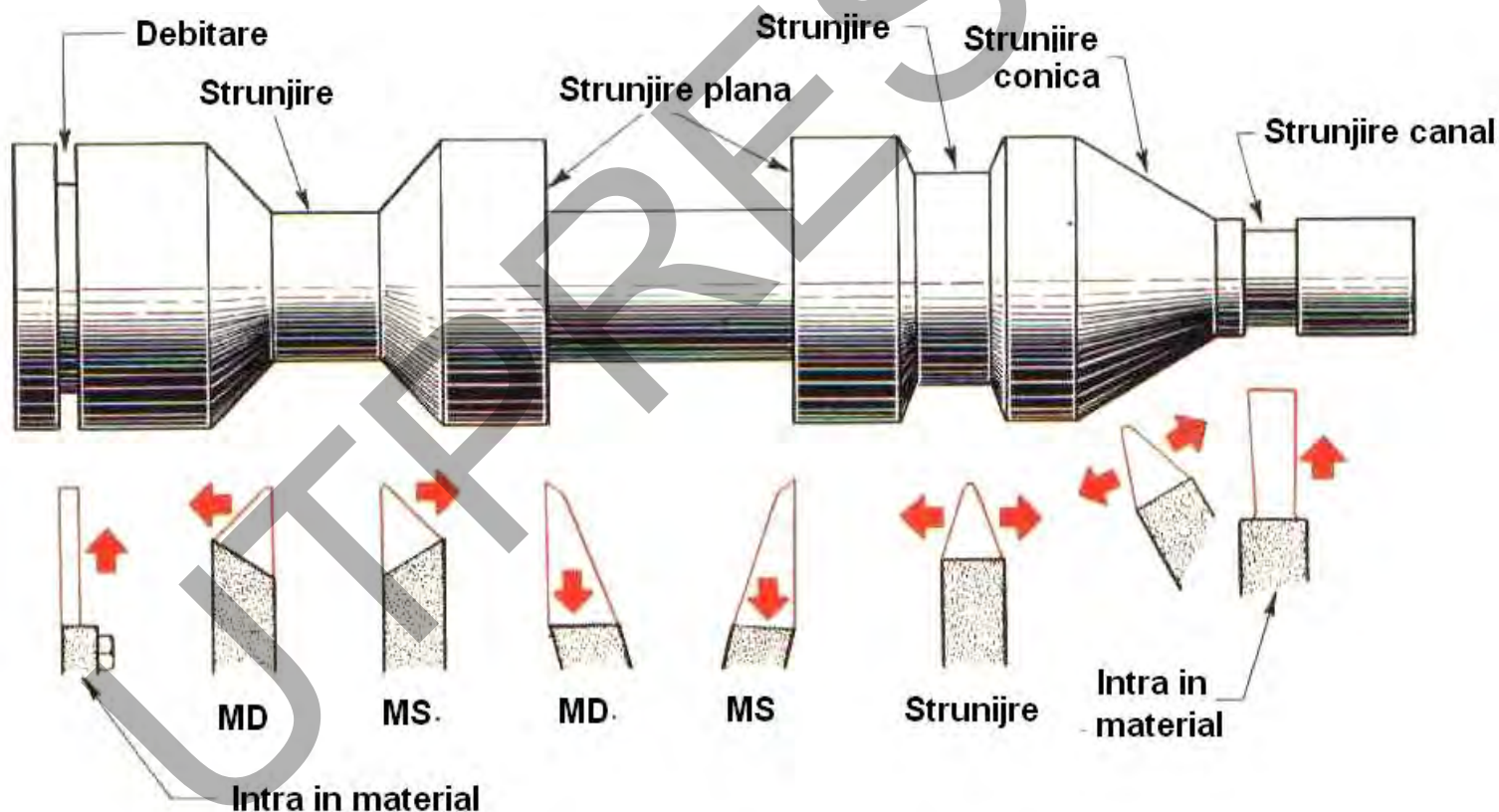
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Scule pe stânga: așchiază de la universal spre păpușa mobilă.

Scule pe dreapta: așchiază de la păpușa mobilă spre universal

Există și scule simetrice.





Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Observații privind deplasarea sculei în punctul de referință: HOME POSITION

- G28 este utilizat pentru programarea deplasării sculei în punctul de referință.
- G28 cere ca în deplasare să se utilizeze un punct intermediar (majoritatea echipamentelor).
- Înainte de programarea codului G28, scula se poziționează într-o zonă de siguranță, ca urmare nu mai este necesară utilizarea punctului intermediar.
- Pentru a indica totuși un punct intermediar se programează o deplasare incrementală de valoare 0:

G28 U0- pe axa X

G28 W0-pe axa Z.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Structura programului este identică cu aceea prezentată la frezare:

1. Program start
2. Schimbare sculă
3. Sfârșit program
4. Funcții pentru prelucrare

**Notă!**

Primele trei sunt aceleași pentru o anumită mașină, dar pot fi diferite pentru diferite mașini.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Particularizări pentru centrul de strunjire TX8D

Program start

%	- început program
0 9999	- număr program, de la 0 la 9999 precedat de 0 (litera)
G21,G40,G95	- condiții inițiale
G28 UO	- deplasare în punctul de referință, axa X
G29 WO	- deplasare în punctul de referință, axa Z
T0202	- încarcă scula #2 cu offsetul #2
G59	- încarcă offsetul punctului de zero piesă
S2000 M3	- pornește AP turația 2000 rot/min, sens direct
G00 Z...	- deplasare rapidă la punctual de start, axa Z
G00 X..	- deplasare rapidă

Este posibilă programarea ambelor axe în același bloc.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

## Schimbare sculă

M9	-Oprire lichid de aşchiere
G27	- Deplasare rapidă în punctul de schimbare sculă, eventual.G28U0W0
T0303	-Încarcă scula #3, offset #3
G59 X0 Z...	-Punctul de zero piesă
S3500 M3	-Turația AP 3500 rot/min, sens orar (direct)
G0 Z..	-Deplasare rapid a pe axa Z la punctul de start.
M8	-Pornire lichid de aşchiere.

## Sfârșit program

M9	oprire lichid de aşchiere
M5	oprire AP
G28 UO WO	deplasare în punctul de referință
M30	sfârșit program
%	sfârșit fișier

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

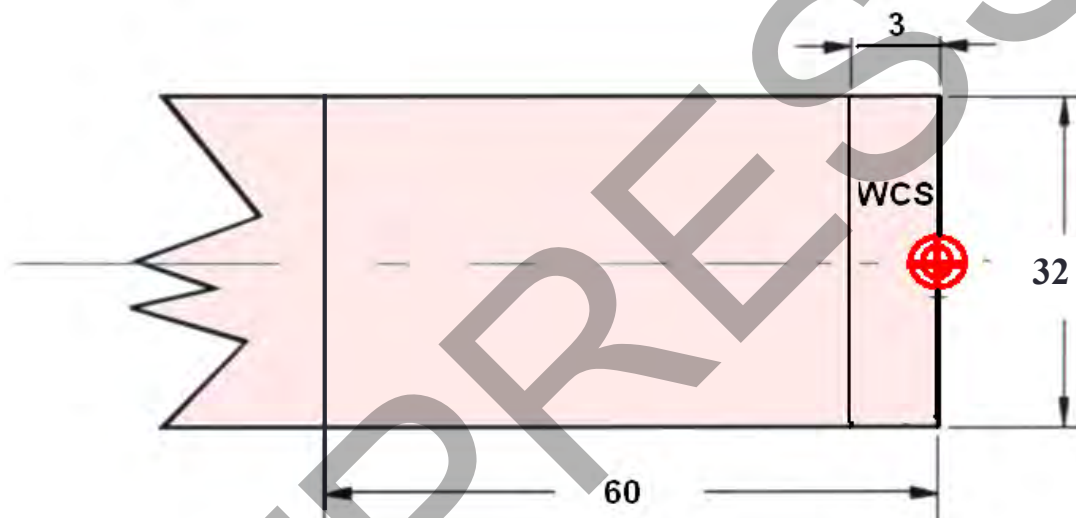
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Exemplu de program de strunjire.

Se cere:

- Prelucrarea frontala a capatului din dreapta pe lungimea de 3 mm



Material :otel  $\sigma_r = 48daN/mm^2$

Obs.: Parametrii regimului de aschiere se calculeaza in acelasi fel ca si pentru operatia de frezare.

- Avansul se exprima (uzual) in mm/rot.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

## Planificare și programare (1-3)

1. **Examinează desenul**
2. **Modul de fixare a materialului: bucașa elastică**
3. **Scula utilizată: cuțit pe dreaptă, plăcuță ceramică, tip C.**

Obs. Se alege viteza de 150 m/min și avansul de 0,1 mm/rot.

Rezultă turația: 1591 rot/min.

Se programează viteza constantă de strunjire G96 V150

## Planificare și programare (4)

### 4. Indicați secvențele de prelucrare

- A. Deplasarea rapidă a sculei, axa Z
- B. Poziționare rapidă a sculei față de piesa la 2 mm, axa X
- C. Strunjire la X0 cu avansul F0.1
- D. Deplasarea rapidă pe axa Z la Z-1
- E. Retragere rapidă pe axa X cu 2 mm
- F. Sfârșit program.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

## Planificarea și programarea (5)

### 5.Convertirea secventelor în programul sursă:

Start program

Strunjire frontală

Sfârșit program

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Programul sursă

%

O 4152

G21 G40 G95

G28 U0 W0

G59 X0 Z190

Se consideră "distanța" MCS - suprafața frontală  
a buclei elastice de 130 mm

G96 V150 TO2O2 M3

G0 X32.0 Z-3.0

G1 X0 F0.1

G0 Z-1.0

G0 X32.0

M9

M5

G28 U0 W0

M30

%



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

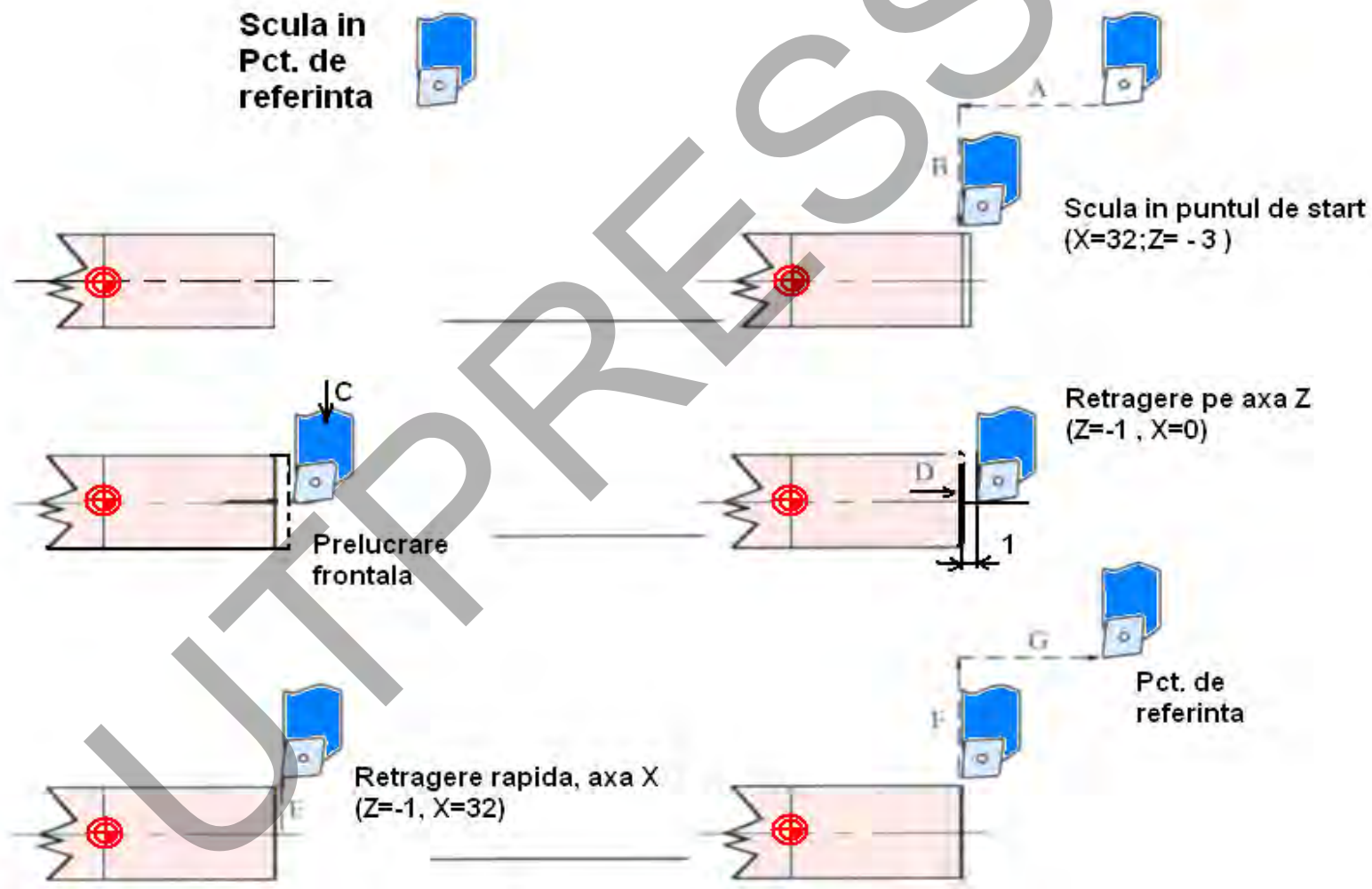
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Ce face mașina?



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Strunjire. Aplicația 2

Cicluri dreptunghiulare



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Animație



Animația reprezintă o operație uzuală de strunjire:

- Deplasare rapidă la diametrul necesar
- Strunjire pe lungimea necesară
- Retragere rapidă pe axa X
- Revenire în punctul de start

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

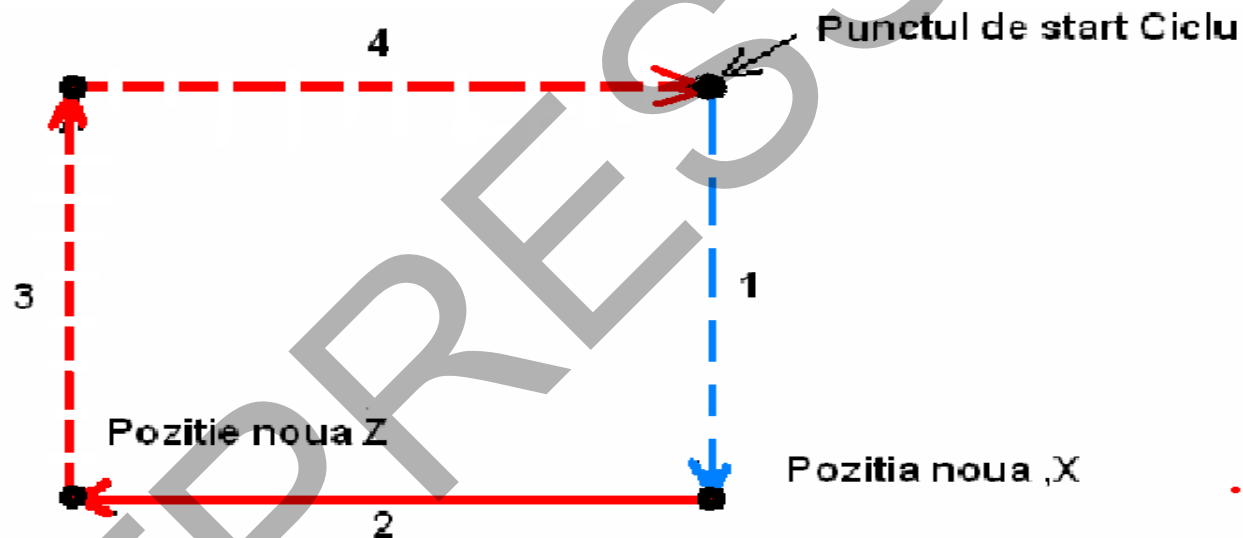
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Ciclul G90

Unele echipamente CNC au implementat, pentru secvența anterioară, un ciclu special (G90).



Programarea:

G90 Xmm Zmm Fmm

- Activarea ciclului se face după poziționarea sculei în punctul de start ciclu.
- Are caracter modal.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

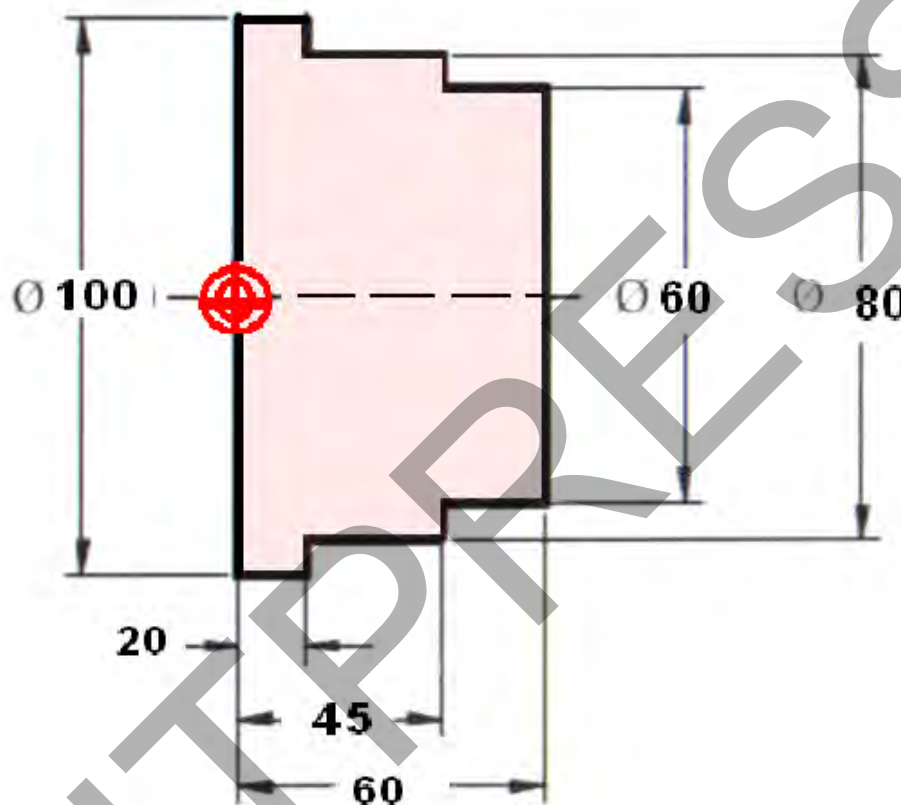
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Ciclul G90-strunjire



Material: Otel carbon  $\sigma_r = 48 \text{ daN/mm}^2$

Semifabricat:  $\varnothing 100 \times 65$

Scula: cu placuta ceramica, tip C, postul 2

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Planificarea și programarea (1-5)

1. Examinare desen
2. Fixare semifabricat: universal cu trei bacuri.
3. Scula utilizată: cu plăcuță ceramică, de tip C, viteza de așchiere 200m/min (constantă)
4. Secvențele de prelucrare
  - Deplasare rapidă la  $z=64$  mm
    - Deplasare rapidă la  $X=104$  mm
    - Așchiere, adâncimea la 90mm și  $Z=0$  (Ciclu dreptunghiular)
    - Sfârșit program
5. Convertirea secvențelor în programul sursă

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Planificarea și programarea (1-5 )

1. Examinare desen
2. Fixare semifabricat: universal cu trei bacuri.
3. Scula utilizată: cu plăcuța ceramică, de tip C, viteza de aşchiere 200m/min (constantă)
4. Secvențele de prelucrare
  - Deplasare rapidă la  $z=64$  mm
  - Deplasare rapidă la  $X=104$  mm
  - Aşchiere, adâncimea la 90mm și  $Z=0$  (Ciclu dreptunghiular)
  - Sfârșit program
5. Convertirea secvențelor în programul sursă



Cap.1. Introducere
Cap.2. Arhitectura unui program
Cap.3. Limbaj ISO
Cap.4. Programarea deplasărilor
Cap.5. Informații tehnologice
Cap.6. Cicluri fixe
Cap.7. Subprograme
Cap.8. Progra marea strungurilor

%  
O4500  
G21 G40 G95  
G28 U0  
G28 W0  
T0202  
G54  
G96 S200 M3 – programare așchiere cu  $V=ct$ ,  $v=200$  mm/min  
G0 Z64.0  
G0 104.0 M8  
G90 X90.0 Z20.0 F0.1  
X80.0 Z20.0  
X70.0 Z45.0  
X60.0  
M9  
M5  
G28 U0 W0  
M30  
%

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

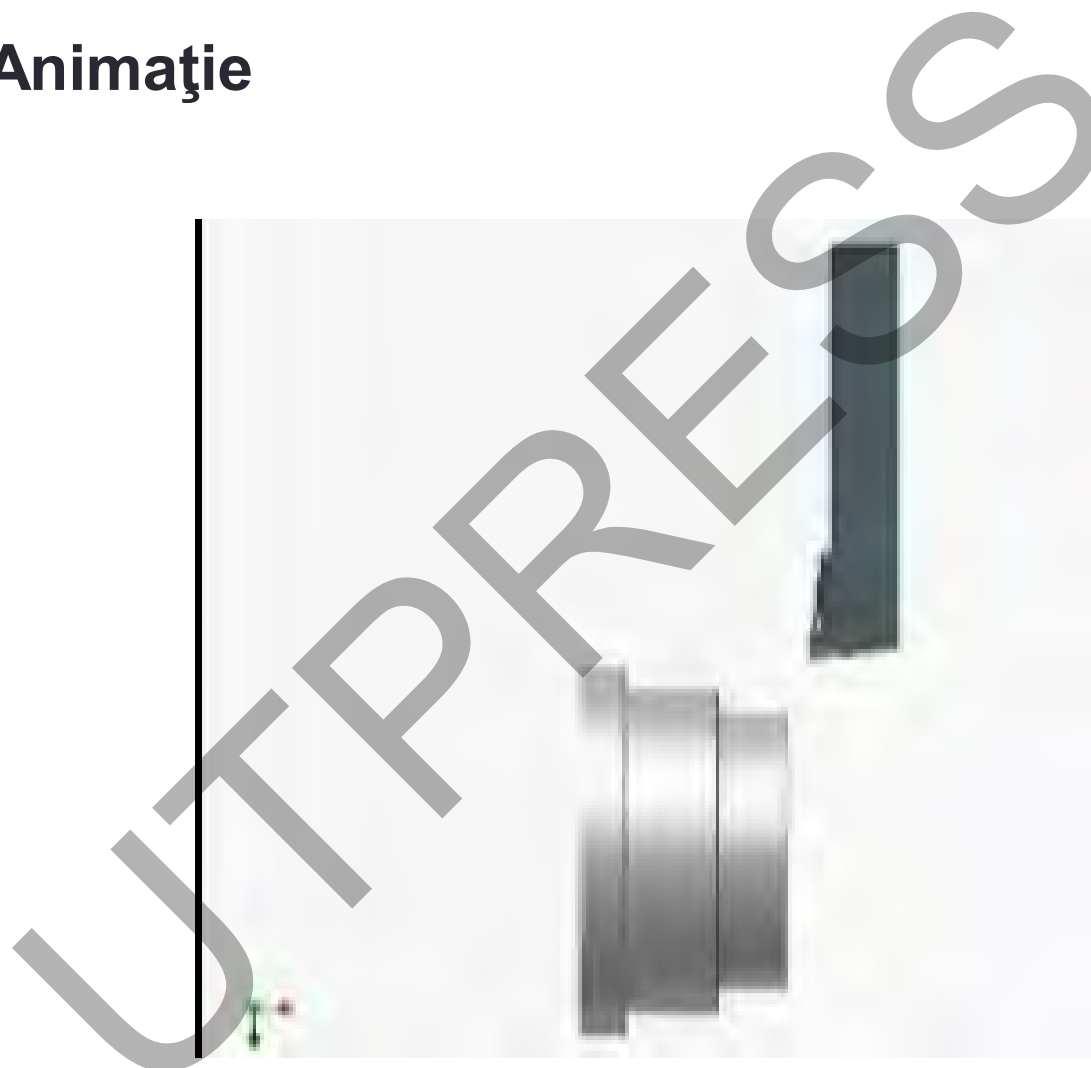
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

# Animație



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Strunjire. Aplicația 3 Filetare



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

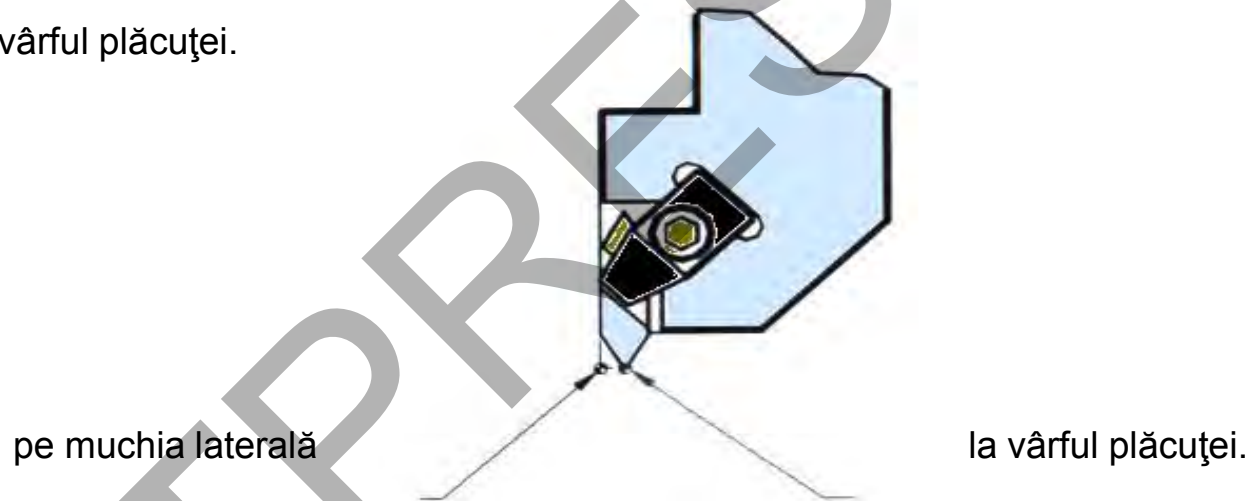
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Offsetul sculelor pentru filetare

Offsetul pentru cuțitul de filetare poate fi stabilit pentru:

- muchia laterală
- vârful plăcuței.



Offsetul programat pentru muchia laterală poate preveni contactul dintre scula și umărul de pe piesă, dar poate “scurta” lungimea filetului.

Offsetul la vârful plăcuței asigură lungimea corectă a filetului însă crește riscul coleziunii sculei cu umărul piesei.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

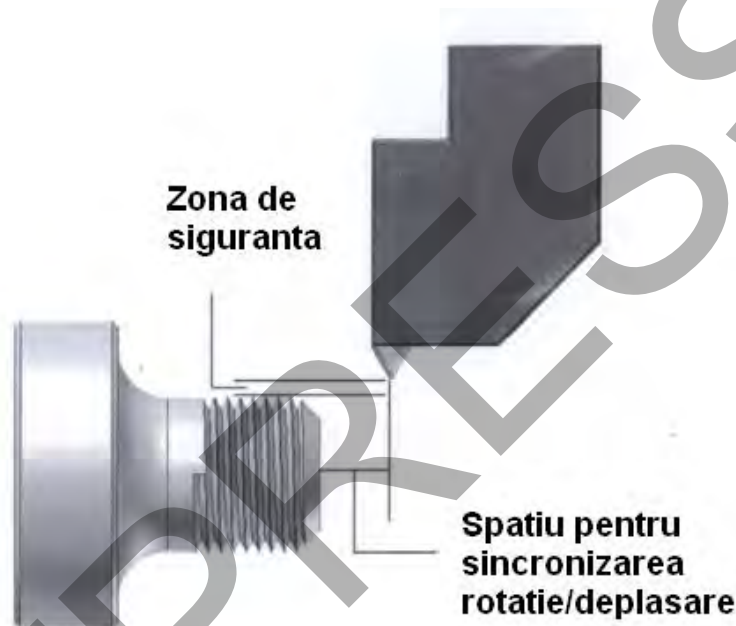
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Zona de siguranță



Trebuie să existe suficient spațiu între scula și piesa pentru a se putea sincroniza deplasarea longitudinală cu turația și pentru a evita o posibilă coleziune.

Obs. dacă se utilizează păpușa mobilă este necesar să se preîntâmpine și coleziunea cu ea.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

## Filet pe dreapta/stânga



Filet pe dreapta:

Turația piesei este în sens invers acelor de ceasornic (direct). M3 deplasarea cuțitului spre păpușa fixă.

Filet pe stânga:

Turația piesei este în sensul acelor de ceasornic iar deplasarea cuțitului se face tot spre păpușa fixă.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

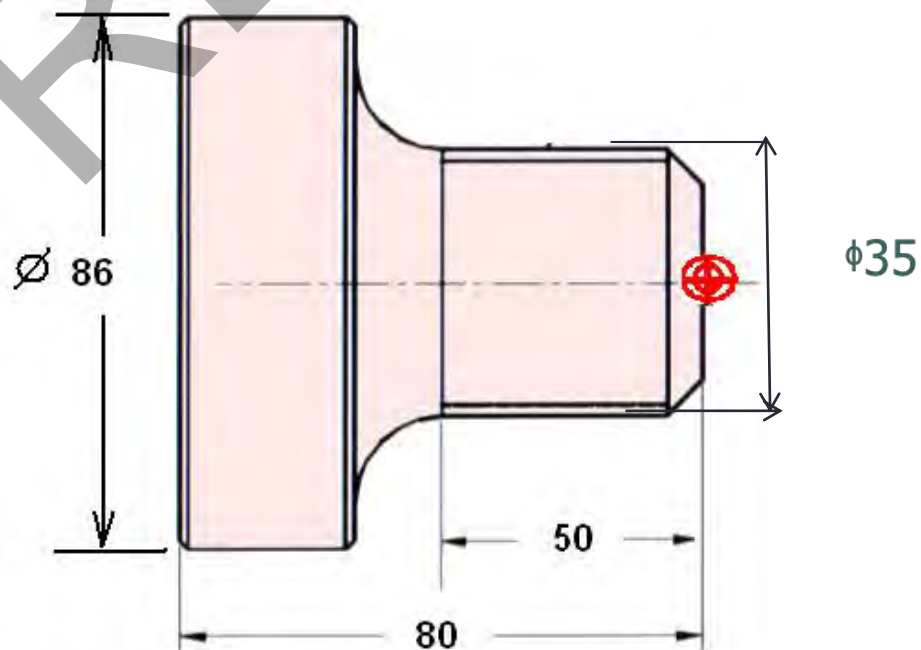
Cap.8. Progra  
marea  
strungurilor

## Exemplu:

Pentru prelucrarea filetului se vor utiliza două cicluri de filetare:

G33-presupune prelucrarea din mai multe treceri

G76- prelucrarea se realizează dintr-o singură trecere



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

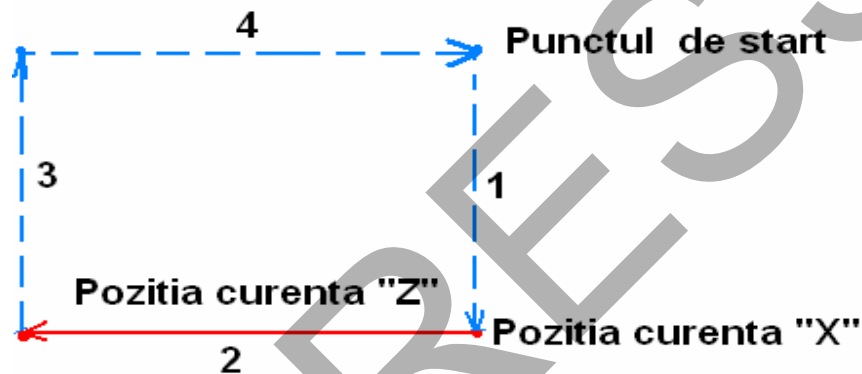
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Ciclul G33

Pentru fiecare trecere de filetare se programează G33. Revenirea sculei se programează separat.



Diametrele trecerilor de filetare:

prima trecere:  $35 - 0,15 \times 2 = 34,70$

a doua trecere:  $34,7 - 0,3 = 34,40$

a treia trecere:  $34,4 - 0,3 = 34,10$

a patra trecere:  $34,1 - 0,3 = 33,80$

a cincea trecere:  $33,8 - 0,3 = 33,5$

a șasea trecere:  $33,5 - 0,3 = 33,2$

Caracteristicile filetelui:

$p = 1,5$

$d = 35$

$d_1 = 33,376$

Înălțimea:  $(d - d_1) / 2 = 0,812$

Numărul de treceri de filetare - 6,  
dispuse astfel:

$5 \times 0,15 = 0,750$

$1 \times 0,062 = 0,062 / 0,812$



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

## Program sursă, G33

%

**O 4500**

**G21 G40 G95**

**Condiții inițiale**

**G28 U0**

**Deplasare în punctul de referință**

**G28 W0**

**T0505**

**Încărcare sculă**

**G54**

**G97 S1500 M3**

**G0 Z4.0 X35.0**

**Poziționare în punctul de start**

**G33 X 34.70 Z -50.0 F1.5**

**G0 X35.0**

**G0 Z4.0.**

.

**G33.2 Z-50.0 F1.5**

**Ultima trecere de filetare**

**G0 X35.0 Z4.0**

**M9**

**M5**

**G28 U0 W0**

**M30**

%

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

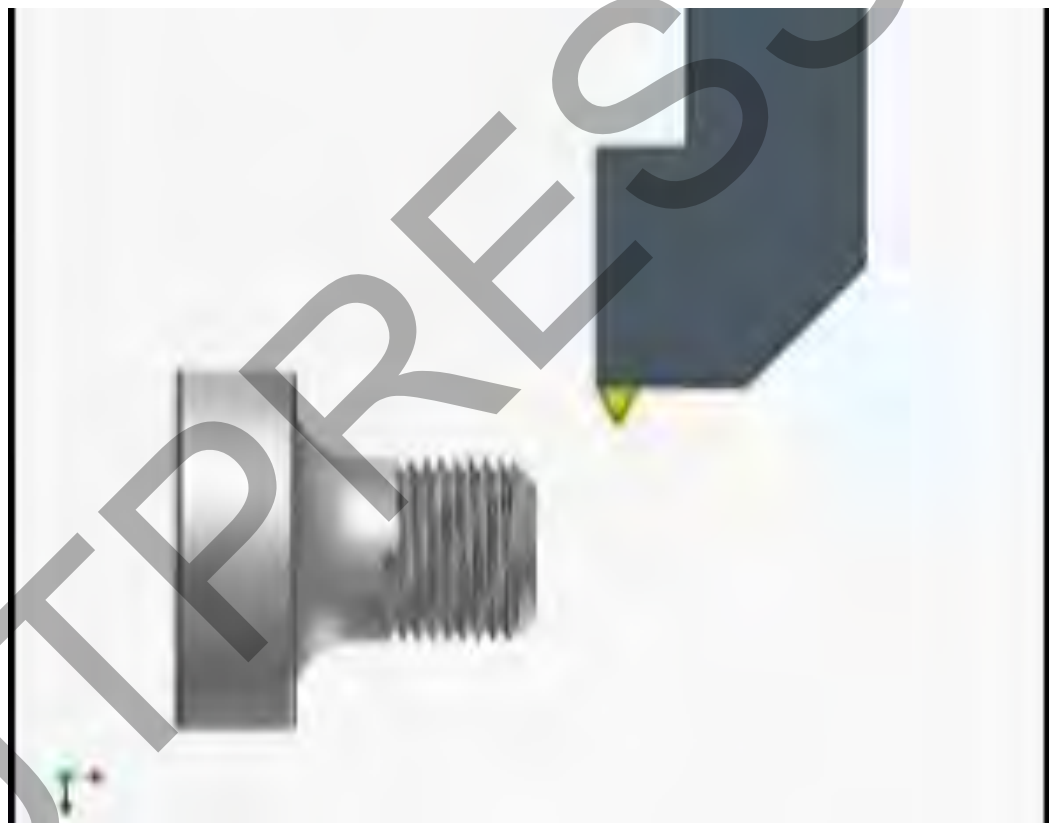
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Animație



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

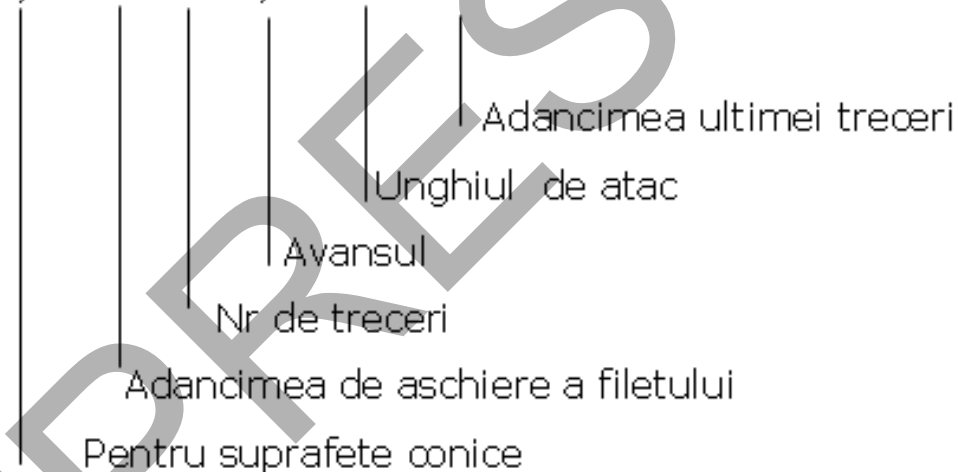
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

## Ciclul G76

Structura blocului de programare

G76 X/U...Z/W...I/J...K...H...F/E...A.....D.....



Blocul pentru programarea ciclului de filetare:

.

.

.

G0 X35.0 Z4.0

G76 X33.2 Z-50 K0.9 H7 F1.5 A20 D0.03

.

.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

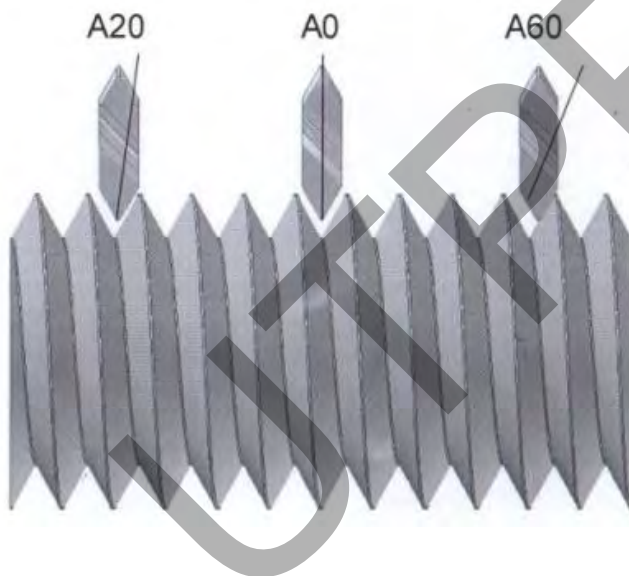
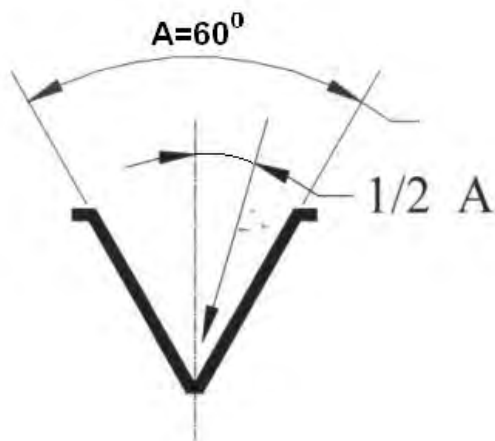
Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor



Codul A?

Filetele metrice (și nu numai) au unghiul de 60. Prin schimbarea unghiului A se poate schimba direcția de pătrundere a sculei.

- Unghiul de pătrundere este  $\frac{1}{2}$  din valoarea specifică sub adresa A din ciclul G76.

A20: așchierea are loc pe muchia principală, cu o finisare bună și are tendința de a minimiza vibrațiile.

A60: toată așchierea este realizată de muchia principală, cu încărcare ușoară a sculei, finisare slabă pentru flancul aferent taisului secundar. Este specific filetării convenționale (manuale).

A0: plăcuța taie cu ambele flancuri, încarcă scula, finisare bună pe ambele flancuri.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

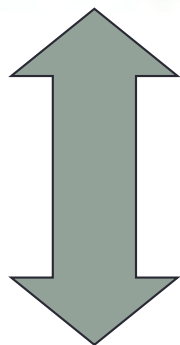
Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## Cap. 9. COMPUTER AIDED MANUFACTURING

**CAM** = Computer Aided Manufacturing  
Computer Automated Machining  
(Prelucrare Asistată de Calculator)



**CNC** = Computer Numerical Control  
(Comandă Numerică Computerizată)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## Obiective

- ✓ Să înțeleagă necesitatea CD/CAM în fabricația asistată de calculator
- ✓ Să poată crea modele geometrice 2D/3D în SPRUT CAM
- ✓ Să poată crea traiectoria centrului sculei în SPRUT CAM
- ✓ Să utilizeze opțiunile din SPRUT CAM pentru stabilirea traiectoriei optime a centrului sculei
- ✓ Să genereze programul sursă de prelucrare pe echipamentul numeric

# Aplicații CAM

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Prelucrarea unor piese cu forme complexe de tipul carcaselor;  
Prelucrarea matrițelor;  
Prelucrarea unor modele complexe pentru turnătorii;  
Prelucrarea unor piese complexe în industria automobilelor;  
Prelucrarea unor came plane și/sau spațiale cu profile complexe;  
Prelucrări de tip Rapid Prototyping;  
Prelucrarea unor piese complexe în industria aerospațială;  
Prelucrarea unor piese din industria lemnului și a instrumentelor muzicale din lemn;  
Prelucrarea unor piese din industria bijuteriilor;  
Practic se pot programa realizarea prin operații de frezare și găurire (iar în ultimele variante și strunjire și debitare) a oricărui tip de piesă definită de suprafețe și/sau curbe analitice sau neanalitice (curbe spline).

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

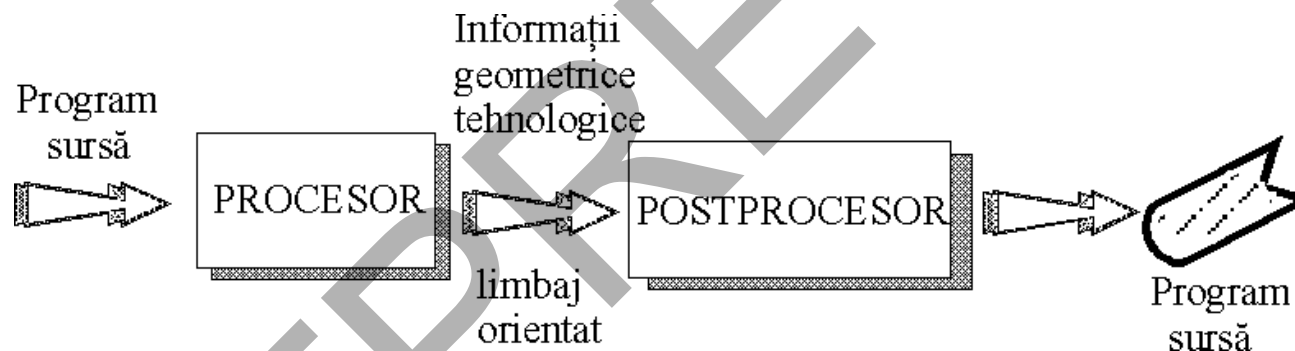
Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

# Aspecte generale

Prelucrarea programului sursă în cadrul calculatorului cuprinde două etape



Calcululele necesare determinării traiectoriei centrului sculei sunt efectuate în etapa PROCESSOR. Limbajul A.P.T. (Automatically Programmed Tool) este cel mai comprehensiv și răspândit limbaj de programare asistată.

Programul sursă conține instrucțiuni prin care programatorul descrie conturul piesei și nu traiectoria sculei.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

În această etapă de prelucrare a datelor în calculator se generează traiectoria centrului sculei, coduri de avans, de turații, etc. Aceste informații sunt depozitate într-un fișier CLDATA (Cutter Location Data) sau CLFILE.

Conținutul fișierului este tradus de un alt program, denumit POSTPROCESSOR, în instrucțiuni codificate, specifice fiecărui echipament NC, sub forma programului de prelucrare.

## Sistemul de programare APT

Sistemul de programare APT a fost dezvoltat de Electronic System Laboratory of the Massachusetts Institute of Technology (MIT) în anul 1956.

Limbajul utilizează caractere:

- alfabetice (literele alfabetului latin, majuscule, 26),
- numerice (cifrele 0,1,2,...9)
- și 13 caractere speciale (; ); +; -; =; \*; \*\*; /; \$\$; \$; .; ;; blanc).

Cu ajutorul caracterelor se formează *cuvintele* – o înșiruire de maximum 6 caractere dintre care primul este obligatoriu de tip alfabetic.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Cuvintele pot fi clasificate în două mari categorii:

- cuvinte rezervate (existente în dicționarul limbajului)
- utilizator (introduse de programator).

Cuvintele rezervate sunt:

- majore (utilizate pentru a defini elemente geometrice, comenzi pentru deplasare, funcții ale mașinii-unelte, indicatori de suprafață, etc)
- minore (în majoritatea cazurilor precizează un anumit element, poziție din entitatea precizată de cuvântul major).

De regulă cuvintele majore sunt plasate în partea stângă a semnului slash „/“ iar cele minore în dreapta lui.

Cuvintele utilizator sunt folosite pentru a defini variabile scalare, variabile geometrice, simboluri geometrice, simboluri de tabele și subprograme, etichete, etc.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

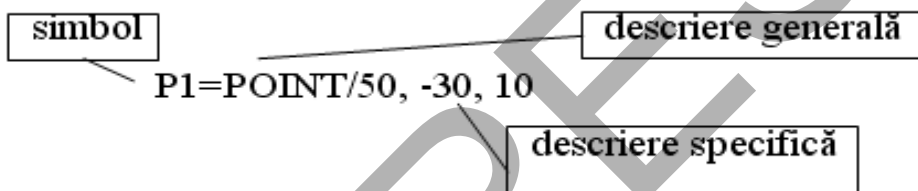
Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Cu ajutorul cuvintelor sunt alcătuite *instrucțiunile* limbajului.

O instrucțiune tipică

-pentru descrierea elementelor geometrice ale piesei este de  
tipul:



- iar pentru comanda deplasării sculei (instrucțiune de mișcare):  
`GOLFT/L1,PAST,L2`  
`GOTO/P1`

Majoritatea instrucțiunilor APT sunt divizate în două secțiuni, majoră și minoră, separate prin semnul slash (/):

-cuvântul GOLFT reprezintă secțiunea majoră a instrucțiunii  
 -cuvântul PAST modificatorul din secțiunea minoră.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbar ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

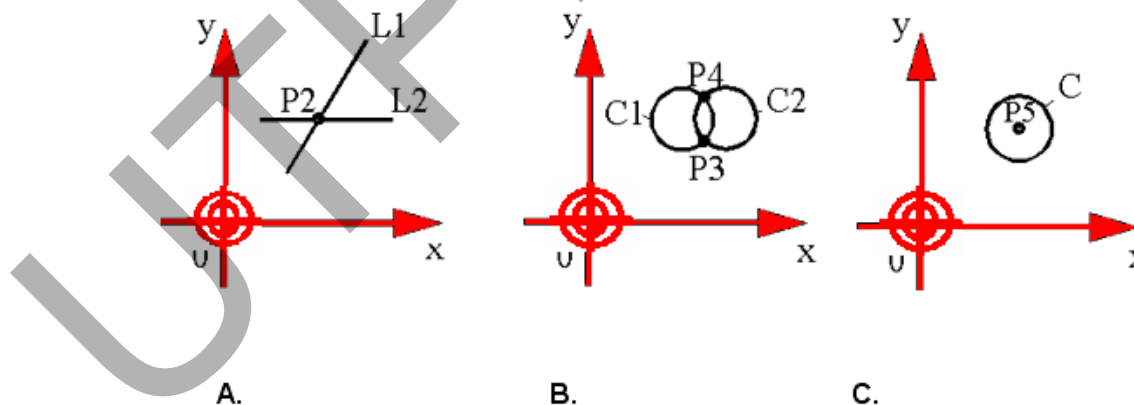
Cap.9. CAM

# Instrucțiuni geometrice

Pentru fiecare instrucțiune geometrică există de la 1 - 14 metode diferite de definire.

Sistemul ATP acoperă definiții pentru 16 elemente geometrice diferite, dintre care cele mai utilizate sunt: POINT, LINE, PLANE, CIRCLE, CYLIND, VECTOR, PATTERN. Cîteva dintre definițiile cele mai uzuale ale acestor elemente sunt prezentate în continuare.

Punctul - poate fi definit în 10 moduri diferite, câteva sunt ilustrate în figura următoare



## Definirea punctului:

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## 1. Prin coordonatele sale

$$P=POINT/abscisă, ordonată [,cotă]$$

$$P1=POINT/20, 40,-10$$

## 2. Prin intersecția a două linii (fig. 8.3.a)

$$P=POINT/INTOF, line- 1, line- 2$$

$$P2=POINT/INTOF, L1, L2$$

## 3. Prin intersecția a două cercuri

$$P = POINT / \left. \begin{array}{l} XSMALL \\ XLARGE \\ YLARGE \\ YSMALL \end{array} \right\} ,INTOF, cerc -1, cerc -2$$

$$P3=POINT/ YSMALL, INTOF, C1, C2$$

$$P4=POINT/ YLARGE, INTOF, C1, C2$$

## 4. Centrul unui arc de cerc

$$P=POINT/CENTER, cerc$$

$$P5=POINT/CENTER, C$$

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

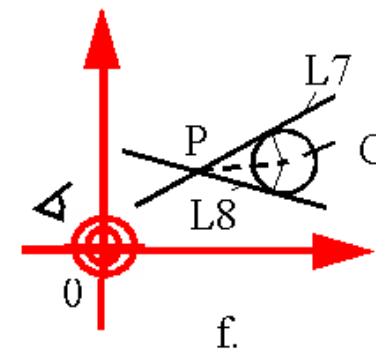
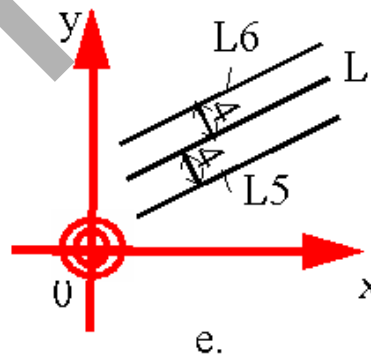
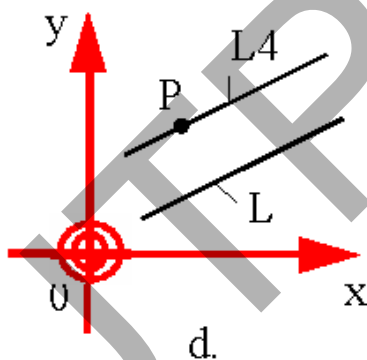
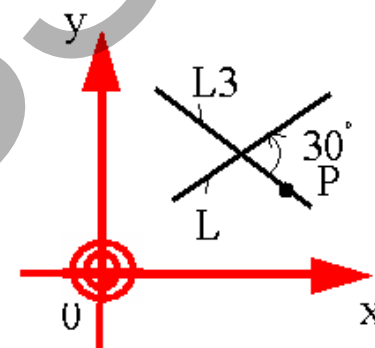
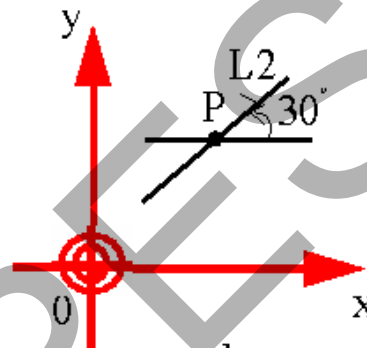
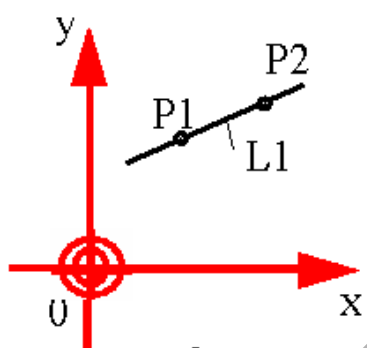
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Linia poate fi definită în 13 modalități diferite.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## EXEMPLE PENTRU DEFINIREA LINIEI

1. Trecând prin două puncte

$L=LINE/ \text{ punct } -1, \text{ punct } -2$

$L1= LINE/ P1, P2$

2. Trecând printr-un punct fiind precizat și unghiul cu axa X

$L=LINE/ \text{ punct}, ATANGL, \text{ valoare unghi}$

$L2=LINE/P, ATANGL, 30$

3. Trecând printr-un punct fiind precizat și unghiul cu o altă dreaptă

$L=LINE/ \text{ punct}, ATANGL, \text{ valoare unghi}, \text{ dreaptă}$

$L3=LINE/ P, ATANGL, 30, L$

4. Trecând printr-un punct și paralelă cu o altă dreaptă

$L4=LINE/ \text{ punct}, PARLEL, \text{ dreaptă}$

$L=LINE/ P, PARLEL, L$

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

5. Paralelă cu o dreaptă și situată la o anumită distanță

$L5 = \text{LINE/PARLEL, L, XLARGE, 4}$

$L6 = \text{LINE/PARLEL, L, YLARGE, 4}$

6. Tangentă la un cerc trecând printr-un punct dat

$L7 = \text{LINE/P, LEFT, TANTO, C}$

$L8 = \text{LINE/P, RIGHT, TANTO, C}$

O definiție similară este aceea pentru cazul dreptei tangente la două cercuri:

$$L = \text{LINE} / \left\{ \begin{array}{l} \text{LEFT} \\ \text{RIGHT} \end{array} \right\}, \text{TANTO, cerc} - 1, \left\{ \begin{array}{l} \text{LEFT} \\ \text{RIGHT} \end{array} \right\}, \text{TANTO, cerc} - 2$$



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

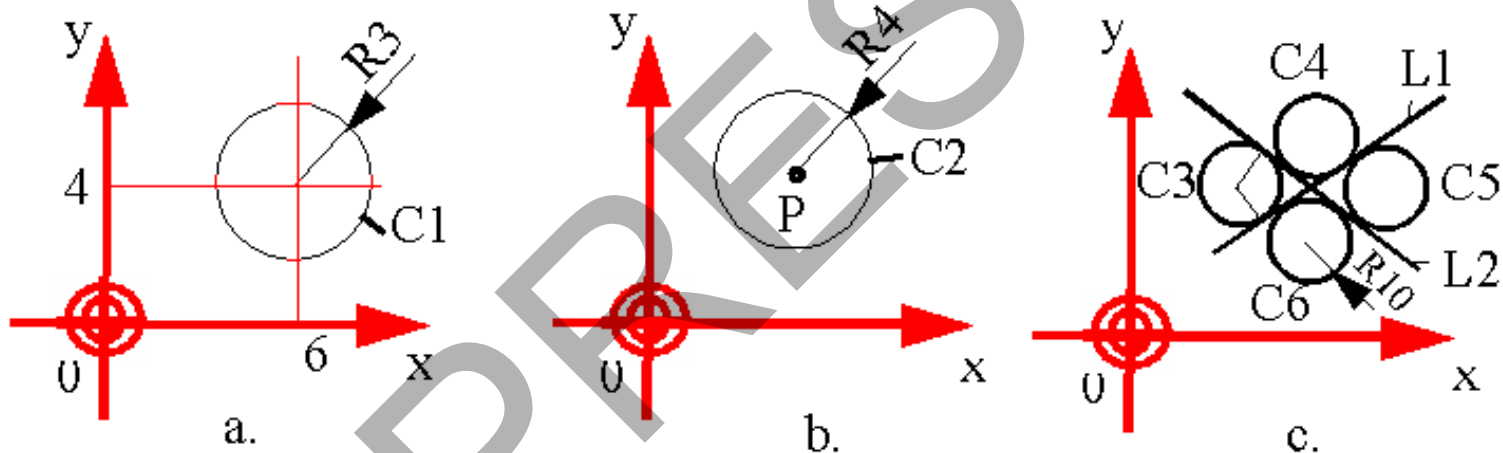
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

*Observație: Modificatorii LEFT, RIGHT precizează poziția dreptei față de cerc privind de la primul element înscris după slash.*



Cercul poate fi definit în 10 moduri diferite

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## EXEMPLE PENTRU DEFINIREA CERCURILOR

### 1. Prin coordonatele centrului și rază

$C=CIRCLE/$  abscisă, ordonată, [cotă], rază

$C1=CIRCLE/$  6.4.3

### 2. Prin centrul său și rază

$C=CIRCLE/CENTER,$  punct,  $RADIUS,$  valoarea razei

$C2=CIRCLE/CENTER,$  P,  $RADIUS,$  4

### 3. Prin rază și tangent la două linii

$$C = CIRCLE / \left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \end{array} \right\}, \text{dreapta} - 1, \left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \end{array} \right\}, \text{dreapta} - 2,$$

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Prin modificatorii XLARGE, XSMALL, etc. se precizează poziția coordonatelor centrului în raport de punctul de tangență cu dreptele.

C3=CIRCLE/YSMALL, L2, XSMALL, L1, RADIUS, 10

C4=CIRCLE/YLARGE, L2, XSMALL, L1, RADIUS, 10

C5=CIRCLE/YLARGE, L2, XLARGE, L1, RADIUS, 10

C6=CIRCLE/YSMALL, L2, XLARGE, L1, RADIUS, 10

Planul poate fi definit prin 8 metode:

1. Prin trei puncte

$PL=PLANE/ punct-1, punct-2, punct-3$

$PL1=PLANE/ P1, P2, P3$

2. Paralel cu alt plan și trecând printr-un punct

$PL=PLANE/punct, PARLEL, plan$

$PL2=PLAN/ P, PARLEL, PL$

3. Paralel cu alt plan, situat la o anumită distanță

$PL = PLANE/PARLEL, plan, \left. \begin{array}{l} ZLARGE \\ ZSMALL \\ XLARGE \\ : \end{array} \right\} dis\ tan\ \breve{t}a$

$PL3=PLANE/ PARLEL, PL, ZLARGE, 50$

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

4. Prin coeficienții A, B, C, D din definirea canonică a planului

$$PL=PLANE/ A, B, C, D$$

$$PL4=PLANE/ 0, 0, 1, 50$$

Planul PL4 este un plan paralel cu planul XOY (A=0, B=0, C=1) situat la distanța Z=50.

Vectorul poate fi definit prin 7 modalități, dintre care se prezintă câteva:

1. Prin componentele după cele trei axe:

$$SVAC=VECTOR/componentă X, componentă Y, componentă Z$$

Direcția și sensul vectorului sunt determinate de mărimea componentelor și de semnul lor:

$$VEC1=VECTOR/10, -20, 30$$

2. Prin punctele extreme ale sale:

$$SVEC=VECTOR/{punct 1, punct 2 Coordonatele lor}$$

$$VEC 2=VECTOR/P1, P2$$

$$VEC 3=VECTOR/10, 20, 30, -40, -50, 70$$

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## Structurile de puncte :

1. Structură liniară definită prin punctele extreme și numărul total de puncte echidistante

*SPAT=PATTERN/LINEAR, punct inițial, punct final, număr de puncte*

*PAT 1=PATTERN/LINEAR, P1, P2, 10*

2. Structură liniară, definită prin punctul de origine, un vector pentru direcționare număr de puncte echispațiate, distanța între puncte

*SPAT=PATTERN/LINEAR, punct, vector, număr, d*

*PAT2=PATTERN/LINEAR, P1, VEC1, 5, 7*

3. Structură liniară, definită prin punctul de origine, un vector pentru direcționare și o succesiune de incremente

*SPAT=PATTERN/LINEAR, punct, vector, INCR, lungime*

*PAT3=PATTERN/LINEAR, P1, VEC1, INCR, 10, 12, 16, 8*

Pentru structurile circulare există definiții asemănătoare cu cele pentru structurile liniare. Sensul de parcurgere a cercului poate fi orar (CLW) sau antiorar (CCLW).

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

4. Structură circulară, definită prin circumferința sa, unghiul de origine și al extremității, sens de rotație și număr de puncte

$$SPAT=PATTERN/ARC, cerc, unghi-1, unghi-2, \begin{Bmatrix} CLW \\ CCLW \end{Bmatrix}, număr$$

$$PAT4=PATTERN/ARC, C, 10, 130, CLW, 7$$

5. Structură circulară, definită prin unghiul inițial, sens de parcurgere și o succesiune de incremente

$$SPAT=PATTERN/ARC, cerc, unghi i, \begin{Bmatrix} CLW \\ CCLW \end{Bmatrix}, INCR, valoare$$

$$PAT5=PATTERN/ARC, C, -40, CCLW, INCR, -10, -20, 30, 40$$

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Instrucțiuni de deplasare

Instrucțiunile de deplasare din sistemul APT permit programarea întregii game de deplasări realizabile pe orice sistem NC. Aceste mișcări pot fi circumscrise următoarelor două categorii:

- de tipul punct cu punct
- de conturare.

Suplimentar trebuie amintită și instrucțiunea de inițializare.

- inițializarea mișcării (precizarea punctului de plecare):

*FROM/ {abscisă, ordonată, [cotă]}, [viteză] {simbol punct}*

Instrucțiuni pentru deplasări de tipul punct cu punct:

*GOTO/ {abscisă, ordonată, [cotă]}, [viteză] {simbol punct}*

pentru deplasări în sistemul absolut și

*GODLTA/ {incr.X, incr.Y, [incr.Z]}, [viteză] {simbol punct}*

pentru sistemul incremental de programare.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Un exemplu de utilizare a acestor instrucțiuni este indicat pentru poziționarea din figura următoare:

FROM/ ST

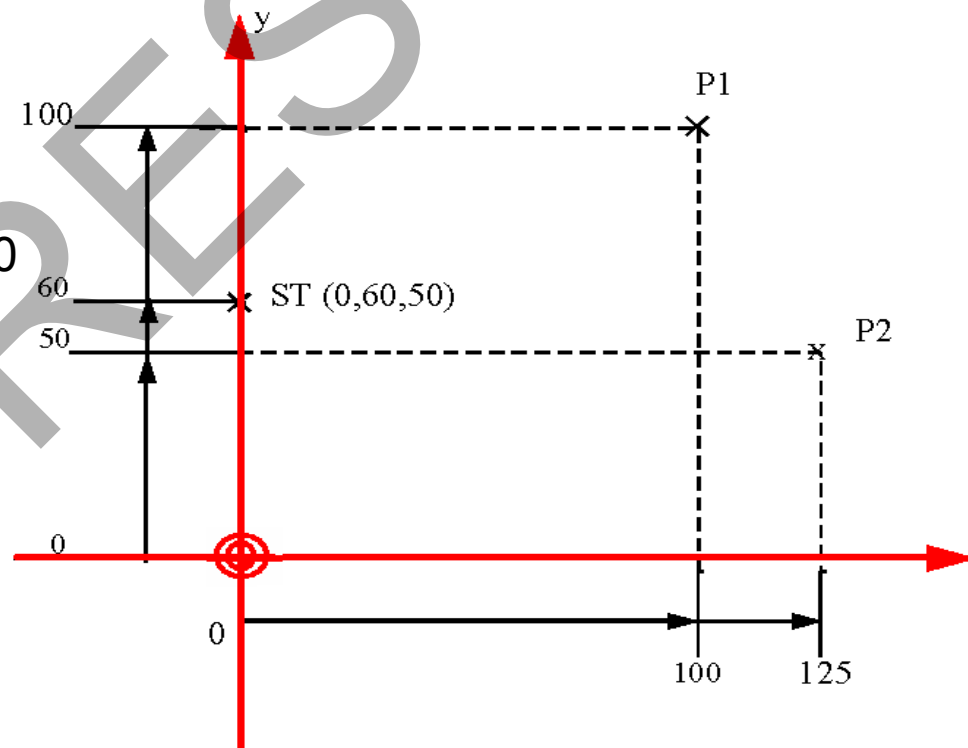
GOTO/ P1

GODLTA/ 0, 0, -60, 50

GODLTA/ 0, 0, 60

GOTO/ P2

:





Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Exemplu pentru utilizarea instrucțiunilor de poziționare

GOTO/ (P1=POINT/ 100, 100, 0)

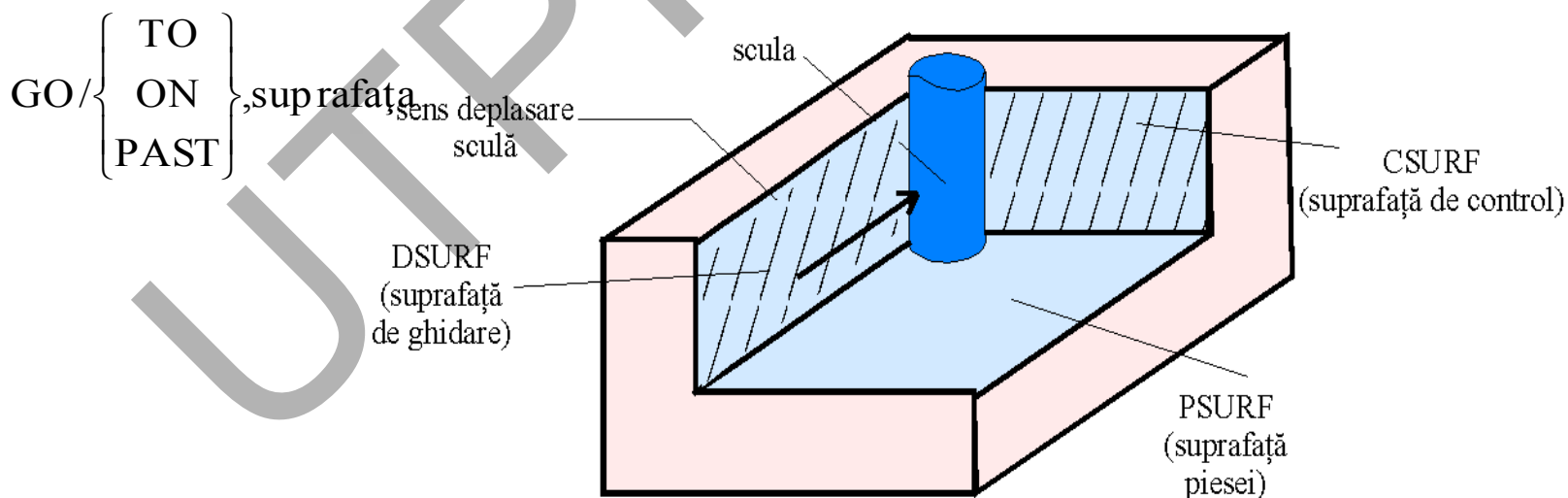
Instrucțiuni de conturare:

- instrucțiuni pentru precizarea direcției de deplasare

*INDIRP/ {abscisa, ordonata, [cota]}{simbol punct}*

*INDIRV/ {vector}*

- instrucțiuni de poziționare a sculei în raport de o anumită suprafață, înainte de intrarea sculei în așchiere.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

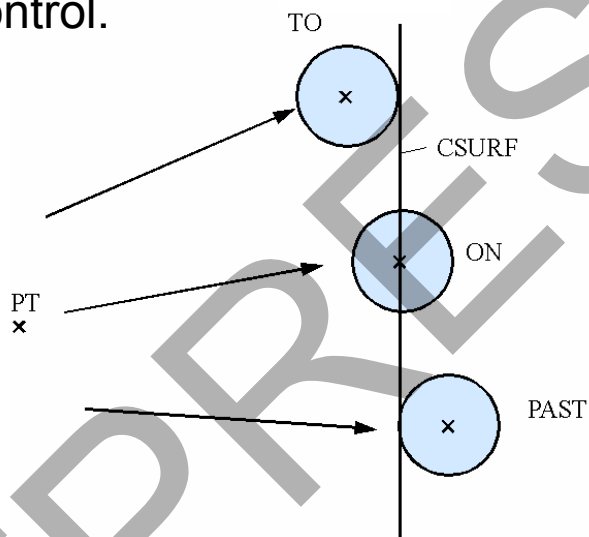
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## Definirea suprafețelor DSURF, CSURF, PSURF

Modificatorii TO, ON, PAST indică poziția finală a sculei în raport de suprafața de control.



- instrucțiuni de deplasare continuă (conturare)

$$\left\{ \begin{array}{l} \text{TLON} \\ \text{TLLFT} \\ \text{TLRGT} \end{array} \right\}, \left\{ \begin{array}{l} \text{TLONPS} \\ \text{TLOFPS} \end{array} \right\}, \left\{ \begin{array}{l} \text{GOLFT} \\ \text{GORGT} \\ \text{GOFWD} \\ \text{GOBACK} \\ \text{GOUP} \\ \text{GODOWN} \end{array} \right\} / \text{suprafața} - 1, \left\{ \begin{array}{l} \text{TO} \\ \text{ON} \\ \text{PAST} \\ \text{TANTO} \end{array} \right\}, \text{suprafața} - 2$$

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

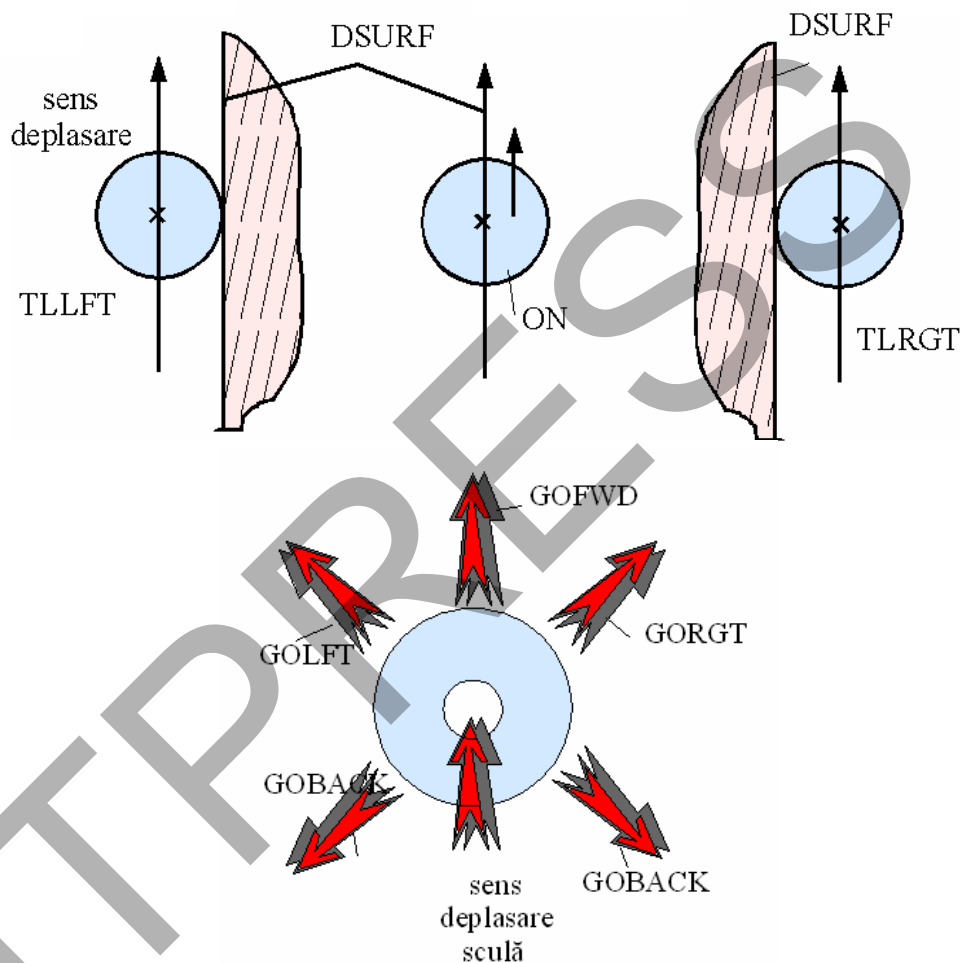
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Suprafața piesei poate fi definită și prin

PSIS/ plan

AUTOPS

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Suprafața de control intersectată de "n" ori:

{...}, {...}, {...}/suprafața-1, {...}, n, INTOF, suprafața-2

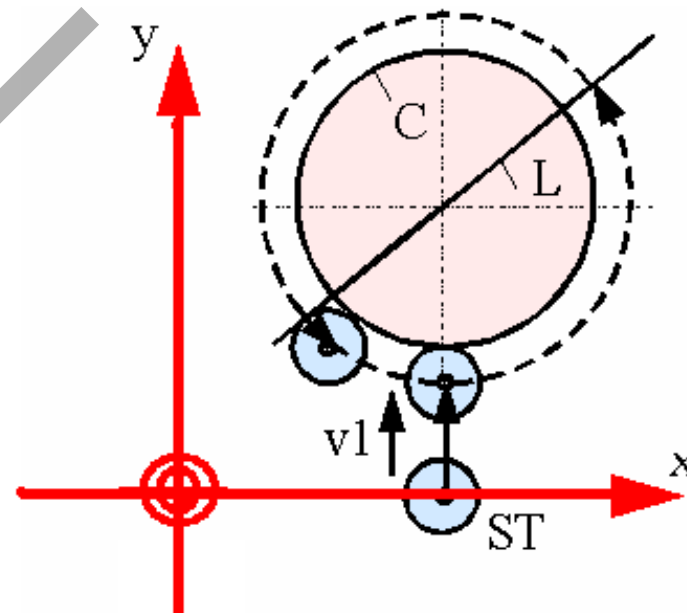
FROM/ ST

INDIRV/ (V1=VECTOR/0,1,0)

GOTO/C

TLRGT, GORGT/C, PAST, 2,  
INTOF, L

FINI



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

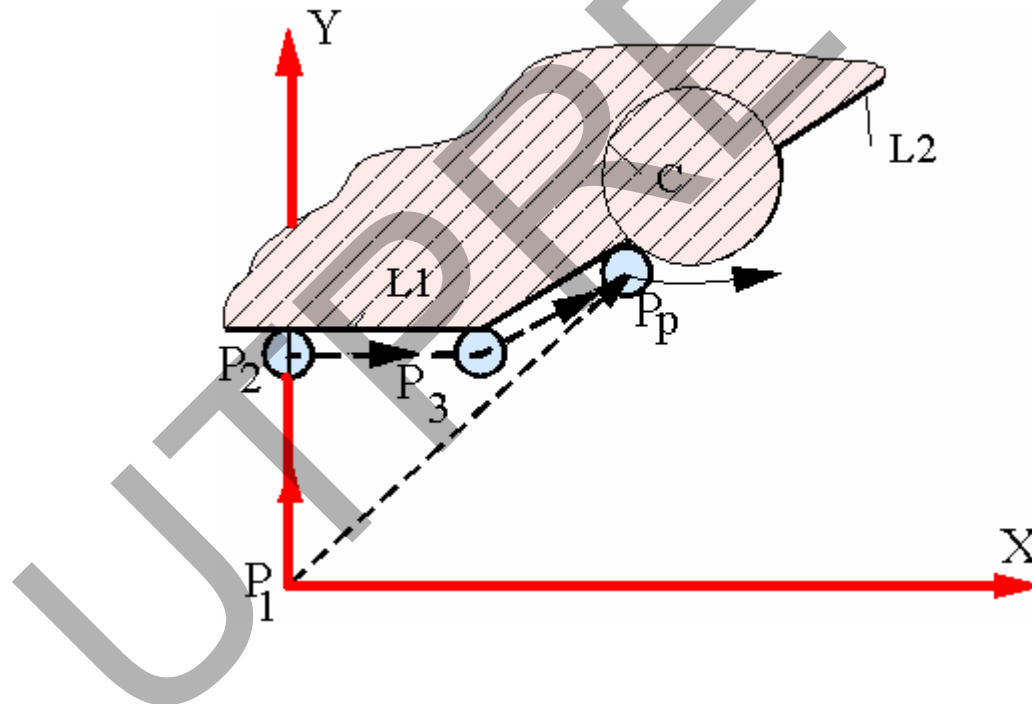
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## POZIȚIONAREA SCULEI ÎN RAPORT CU CSURF

Instrucțiunile CUT și DNTCUT.



## ILUSTRAREA INSTRUCȚIUNILOR DNTCUT ȘI CUT

\$\$ DEFINIRI GEOMETRICE:

L1=LINE/0, 10, 20, 10

L2=LINE/0, 4, 60, 40

C1=CIRCLE/20, 17, 0, 6

\$\$UTILIZAREA INSTRUCȚIUNILOR CUT ȘI DNTCUT

PENTRU DEPLASARE PE DIRECȚIA P1 P2

CUTTER/10

FROM/(P1=POINT/0, 0, 0)

DNTCUT

INDIRP/ (P2=POINT/ 0, 10)

GO/ TO, L1

TLRGT, GORGT/L1, PAST, L2

GOLFT/L2, TO, C

CUT

FINI

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

# Instrucțiuni logice și subprograme

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

- LOOPST, începutul buclei
- LOOPND, sfârșitul buclei.

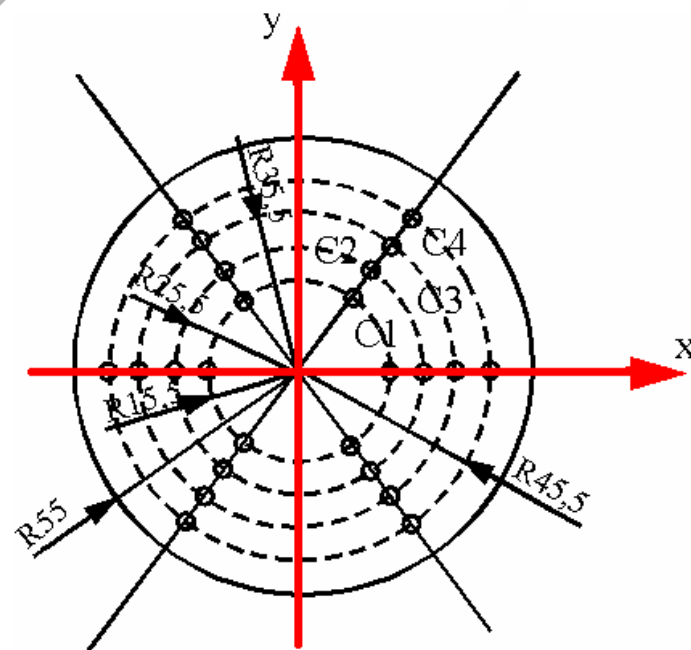
Într-o buclă se pot utiliza două tipuri de instrucțiuni de salt:

- Saltul necondiționat: *JUMPTO/ etichetă*
- Saltul condiționat: *IF (expresie aritmetică) eticheta-1, eticheta-2, eticheta-3*

Exemplu:

```

FROM/0,0,0
R=15.5
LOOPST
1) C=CIRCLE/0,0,0,R
PAT=PATTERN/ ARC, C, 0, 300, CCLW, 6
GOTO/PAT
CYCLE/DRILL
R=R+10
IF (R- 45.5) 2,2,3
2) JUMPTO/1
3) LOOPND
FINI
  
```



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Subprogram este o unitate de program independentă, care poate fi utilizată în același program sau în mai multe programe, definită prin numele său, parametrii de intrare, secvența program și o instrucțiune finală. Există două modalități de a indica formatul de programare:

$$-symbol = MACRO / [A1, A2, \dots, An]$$

în care :

-simbolul este un cuvânt utilizator cu care poate fi apelat subprogramul;

-A1, A2, . . . , An sunt variabile care, în momentul apelării subprogramului, conțin valori ce constituie parametrii subprogramului.

$$-symbol = MACRO / [A1=val-1, A2=val-2, \dots, An=val-n]$$

Instrucțiunea *TERMAC* marchează sfârșitul unui subprogram.

Apelarea subprogramelor se face prin instrucțiunea *CALL*. Formatul general al acestei instrucțiuni este:

$$CALL / symbol \ subprogram, A1=val-1, A2=val-2, \dots, An=val-n$$



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

RESERV/C, 6

MAC1=MACRO/R, I

C(I)=CIRCLE/0,0,0,R

PAT1=PATTERN/ARC, C(I), 0,300, CCLW, 6

GOTO/PAT1

CYCLE/DRILL

TERMAC

CALL/MAC1, R=15.5, I=1

CALL/MAC1, R=25.5, I=2

CALL/MAC1, R=35.5, I=3

CALL/MAC1, R=45.5, I=4

FINI

Instrucțiunea TERMAC este obligatoriu să fie programată la sfârșitul subprogramului .

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Alte instrucțiuni APT

MACHIN/ nume postprocesor

Instrucțiunile APT de tip postprocesor sunt transformate de către postprocesor în coduri pentru echipamentul numeric:

*COOLNT/ ON* - pornire lichid de răcire (M08);  
*SPINDL/ON* - pornire arbore principal (M03);  
*FEDRAT/60* - viteza de avans (F60);  
*END* - sfârșit de program (M02);  
*REWIND* - sfârșit de program cu rebobinare de bandă (M30);  
*RAPID* - deplasare cu avans rapid (G00).  
*OUTTOL/.001* - toleranță exterioară;  
*INTOL/.001* - toleranță interioară;  
*TOLER/.002* - toleranță exterioară și interioară;

În care „valoare“ de după "/" semnifică diametrul sculei.

Instrucțiunea prin care se definește numele programului este:

*PARTNO* nume program.

Ultima instrucțiune dintr-un program APT este cea de sfârșit de program: FINI.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

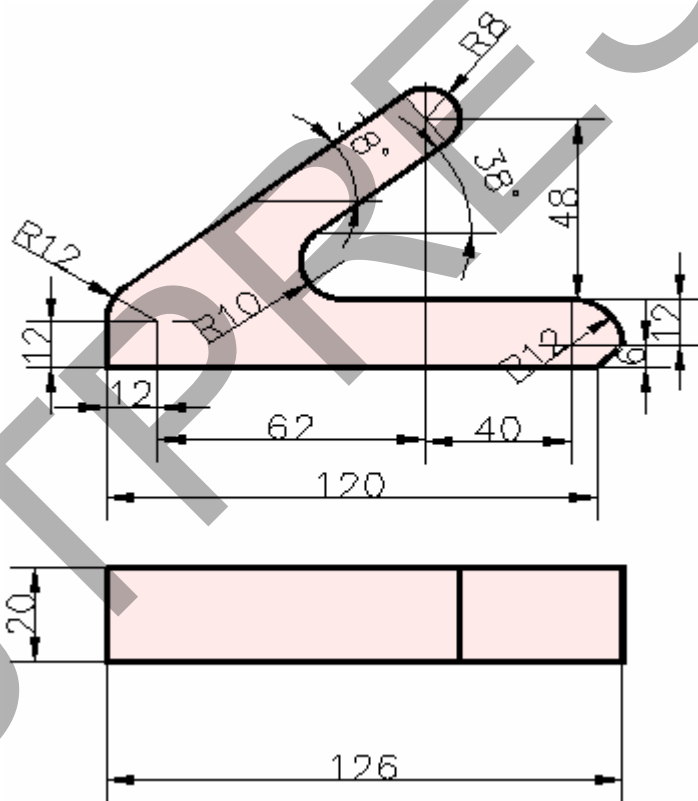
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Exemplu de program APT

Pentru ilustrarea metodei de programare asistată în sistemul APT se consideră piesa



# Piesă model pentru programare în limbajul APT-RCV

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

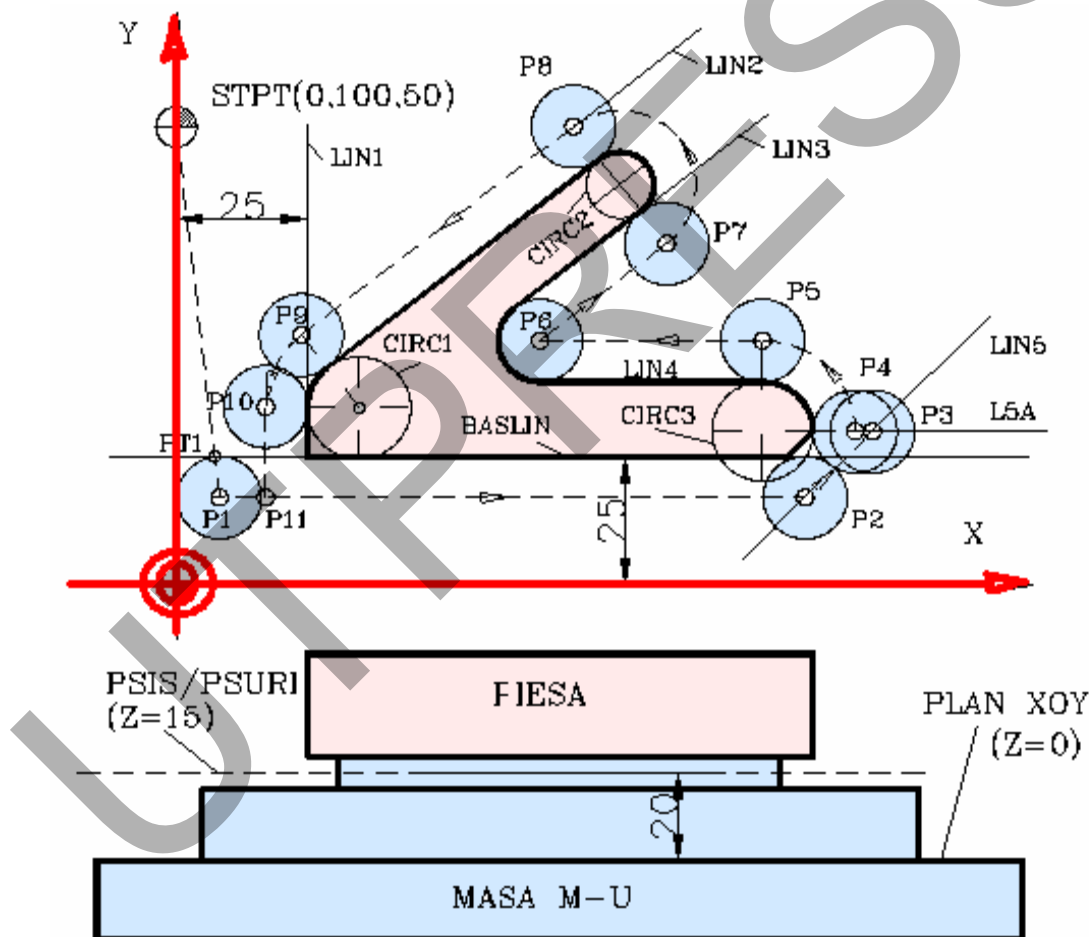
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## PARTNO EXEMPLU DE PROGRAM APT

```

INTOOL/.005
OUTTOL/.005
CUTTER/20
ST          = POINT/0,100,50
PT1        = POINT/5,25
PT2        = POINT/145,25
PT3        = POINT/139,43
BASLIN     = LINE/PT1,PT2
C1         = CIRCLE/ CENTER, (POINT/ 12,12), RADIUS, 12
L1         = LINE/ (POINT/ 25,25), LEFT, TANTO, C1
C2         = CIRCLE/99, 91, 8
L2         = LINE/ LEFT, TANTO, C1, LEFT, TANTO, C2
C3         = CIRCLE/ 139, 31, 12
L4         = LINE/(POINT/99, 43), LEFT, TANTO, C3
L3         = LINE/(POINT/INTOF, L4, (LINE/ PARLEL, L2, XLARGE, 16)), $
           RIGHT, TANTO, C2

```

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## Instrucțiuni de deplasare:

SPINDL/950,CLW

FEDRAT/500

COOLNT/ON

FROM/ST

GODLTA/0,0,-35

AUTOPS

INDIRP/PT1

G0/PAST,BASLIN

FEDRAT/60

TLRGT, GOLFT/BASLIN, PAST, L5

GOLFT/L5, ON (L5A=LINE/10,31,20,31)

TLON, GOLFT/L5A, TO, C3

TLRGT, GORGT/C3, ON, (LINE/PT3, PERPTO, L4)

GOFWD/L4

GORGT/L3

GOFWD/C2

GOFWD/L2

GOFWD/C1

GOFWD/L1, PAST, BASLIN

COOLNT/OFF

FEDRAT/ 500

GODLAT/0,0,35

GOTO/ST

REWIND

FINI

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

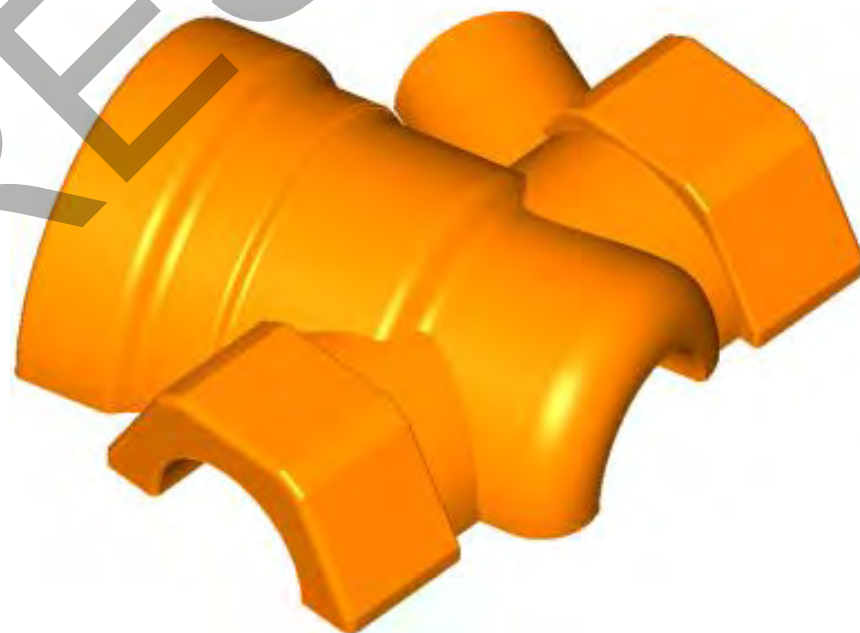
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Generarea modelui geometric al piesei de prelucrat

- ✓ modelare în programe CAD
  - ✓ Alibre Design
  - ✓ CATIA
  - ✓ CoCreate
  - ✓ Inventor
  - ✓ Pro-E
  - ✓ Siemens NX
  - ✓ Solid Edge
  - ✓ SolidWorks
  - ✓ ZW3D



**Modelul 3D al piesei**

- ☐ scanare 3D (reverse engineering)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# PRE-PROCESARE (Calculul parametrilor geometrici și tehnologici)

## Intrări

- tipul prelucrării (frezare, strunjire, etc)
- sculele disponibile
- forțele limită admisibile
- grosimea maximă a așchiei
- limitele mașinii-unelte (mărimea spațiului de lucru)

## Ieșiri

- selectarea sculei
- divizarea traiectoriilor de lucru în entități elementare (segmente de dreaptă, arce de cerc)
- calculul vitezei de avans și adâncimii de așchiere pe traiectoriile elementare
- estimarea vitezei de așchiere



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

# PROCESARE

## (Generarea comenzilor pentru execuția piesei bloc cu bloc)

### Intrări

- tipul frazei (interpolare liniară sau circulară, etc.)
- regimul de aşchiere și geometria detaliată a sculei
- proprietățile materialului de prelucrat

### Ieșiri

- comenzile sub formă de incremente de deplasare
- valorile constrângerilor (forțe, grosimea aşchiei)
- direcția și sensul mișcării
- urmărirea durabilității teoretice a sculei (să nu se depășească această valoare)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

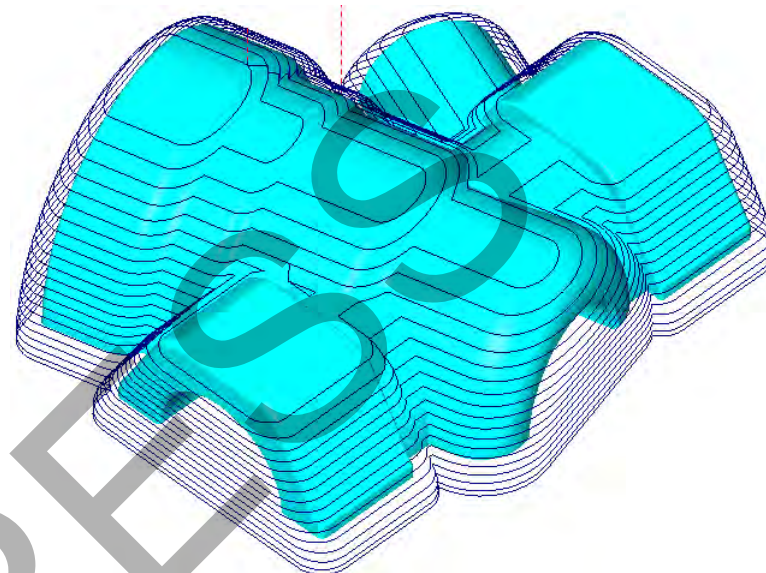
Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

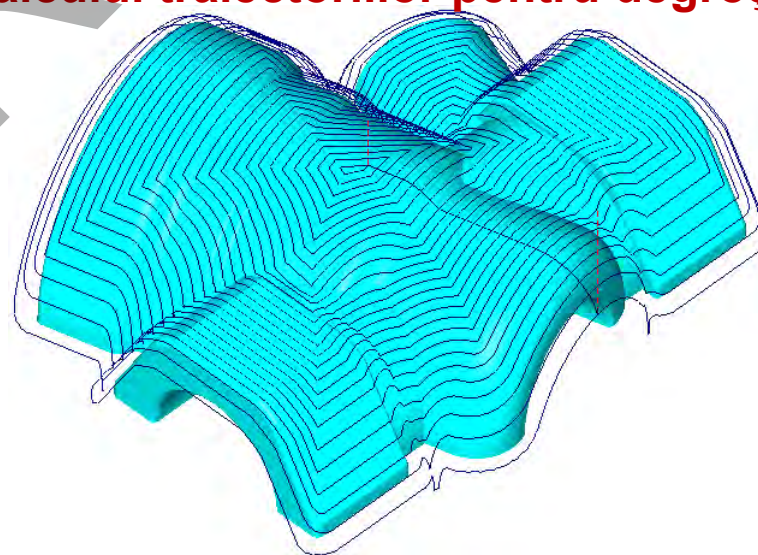
**PRE-PROCESARE**

**+**

**PROCESARE**



**Calculul traiectoriilor pentru degroșare**



**Calculul traiectoriilor pentru finisare**

# POST-PROCESARE

(generarea codului NC specific echipamentului CNC și a comenzilor către axele mașinii și sistemele de control)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

.....  
T3M6  
G54  
S159M3  
G00G43H3X113.449Y57.394Z20.  
Z-6.748  
G01Z-7.748F50  
G02X109.915Y58.862I0.005J5.F100  
G03X96.566Y69.91I-57.923J-56.399F200  
G01X76.379F600  
X88.04Y63.503F200  
G02X109.915Y43.44I-38.401J-63.828  
G01Y22.519F600  
G03X79.931Y56.577I-59.905J-22.512F200  
G01X55.512Y69.91  
G02X53.111Y72.89I2.397J4.389  
G00Z20.  
.....

**Limbaj ISO (cod G și M)**

.....  
33 TOOL CALL 3 Z S159  
34 TOOL DEF 4  
35 L Z+500 FMAX M91  
36 M12  
37 L X+113.449 Y+57.394 A+0 R0 FMAX M3  
38 L Z-6.748 R0 FMAX  
39 L Z-7.748 R0 F50  
40 CC X+113.454 Y+62.394  
41 C X+109.915 Y+58.862 DR- F100  
42 CC X+51.992 Y+2.463  
43 C X+96.566 Y+69.91 DR+ F200  
44 L X+76.379 R0 F600  
45 L X+88.04 Y+63.503 R0 F200  
46 CC X+49.639 Y-0.325  
47 C X+109.915 Y+43.44 DR-  
48 L Y+22.519 R0 F600  
49 CC X+50.01 Y+0.007  
50 C X+79.931 Y+56.577 DR+ F200  
.....

**Limbaj Heidenhain**

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# POST-PROCESARE

(generarea codului NC specific controllerului CNC și a comenzilor către axele mașinii și sistemele de control)



**Execuția degroșării**



**Execuția finisării**

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

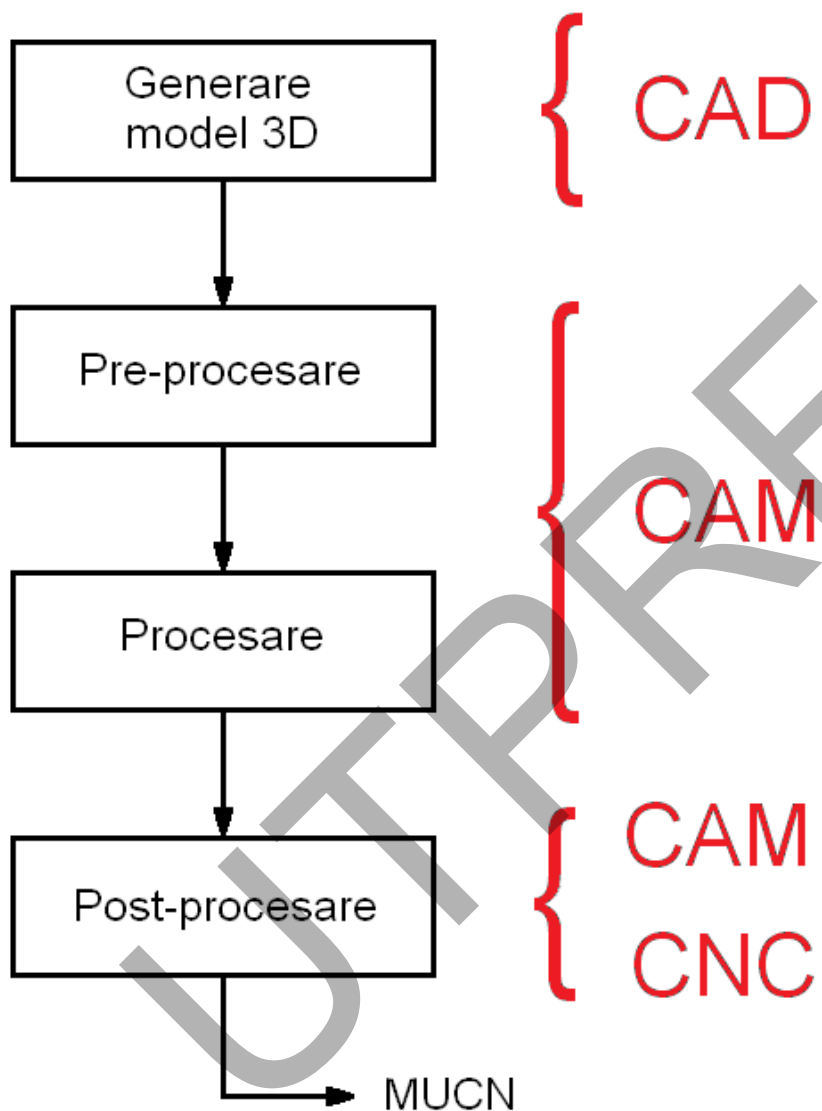
Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Ce este necesar pentru a începe SPRUTCAM

Desenul piesei

- Creat sau importat

Completul de scule

- Selectare din bibliotecă sau create

Planificarea prelucrării

Dispozitive de fixare

- Definiere orientare, locație

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Etape ce trebuie parcurse pentru obținerea codului NC în SprutCAM

## Etapa 1: Reprezentarea grafică a piesei (modelul)

realizarea modelului geometric 3D al piesei într-un program de proiectare/modelare asistată (CAD) extern (introducere directă a modelului);

realizarea modelului geometric al piesei se poate realiza și prin scanare 2D (de pe desen) sau 3D (de pe un model fizic existent al piesei, utilizând scanere speciale);

importul modelului din programul CAD (fișier DXF, IGES sau alte formate suportate);

După cum se poate observa, modelul geometric al piesei se realizează indirect, în afara pachetului SprutCAM, existând însă și posibilități de modelare directă, în program.

La finalul acestei etape se obține un fișier grafic conținând informații complete despre geometria piesei (modelul).

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## Etapa 2: Definirea traiectoriei sculei prin selectarea geometriei de prelucrat

selectare contururi;  
selectare entități de tip insulă/buzunar;  
selectare structuri de găuri;  
selectare suprafețe;  
selectare volume;

La finalul acestei etape programul SprutCAM deține informații despre elementele geometrice care vor fi prelucrate (**răspunsul la întrebarea CE urmează a fi prelucrat ?**).

## Etapa 3: Definirea parametrilor de prelucrare

informații tehnologice legate de sculă (tip, turație, avans);  
definirea formei semifabricatului;  
strategia de prelucrare (prelucrare buzunar tip zigzag, prelucrare spirală, parametrii prelucrărilor de degroșare, parametrii prelucrărilor de finisare, etc.).

La finalul acestei etape programul SprutCAM deține informații despre parametrii de prelucrare (**răspunsul la întrebarea CUM urmează a fi prelucrată piesa?**).



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

#### **Etapa 4: Simularea prelucrării**

se obține o reprezentare grafică a deplasării sculelor și a prelucrărilor realizate;

simulatorul permite definirea și vizualizarea port-sculelor utilizate;

ca rezultat al simulării se obține fișierul CLDATA conținând toate datele legate de deplasarea sculei și a secvențelor de prelucrare;

Simularea permite identificarea eventualelor coliziuni și oferă posibilitatea modificării fișierului de comenzi CLDATA generat în scopul eliminării acestora.

#### **Etapa 5: Modificarea secvențelor de prelucrare**

programul SprutCAM permite realizarea de modificări în fișierul CLDATA

#### **Etapa 6: Post-Procesarea**

conversia fișierului CLDATA în format NC specific unui anumit tip de echipament (programul SprutCAM include o bibliotecă vastă de peste 100 de post-procesoare);

programul permite și realizarea de post-procesoare, în cazul în care acestea nu se regăsesc în biblioteca implicită;

informațiile sunt filtrate și formatate conform codului NC acceptat de echipament, luându-se în considerare parametrii specifici ai mașinii pe care se execută prelucrarea (spațiul de lucru, limitări ale turației și avansurilor, scule, etc.).

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

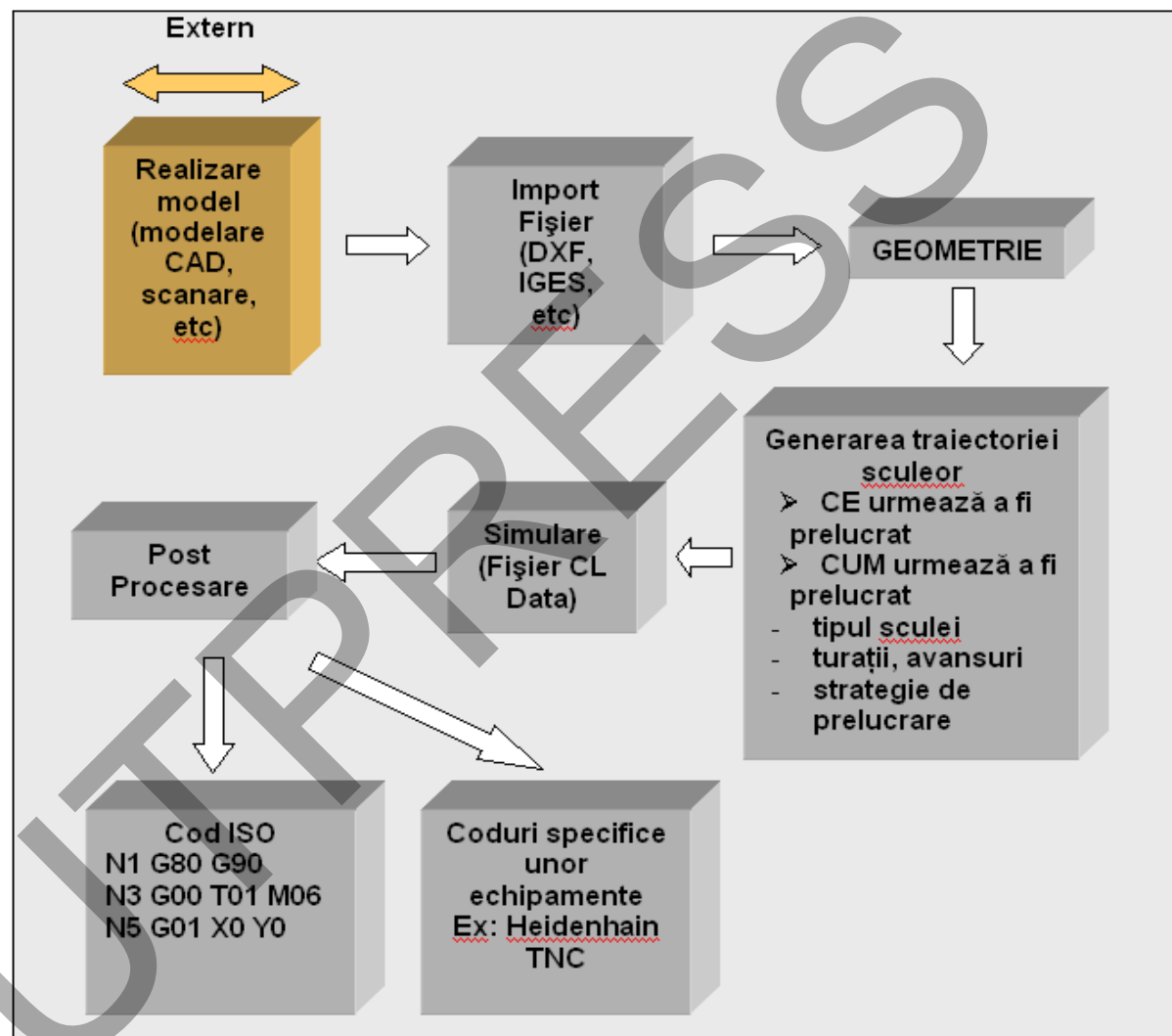
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Exemplu de abordare CAM

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

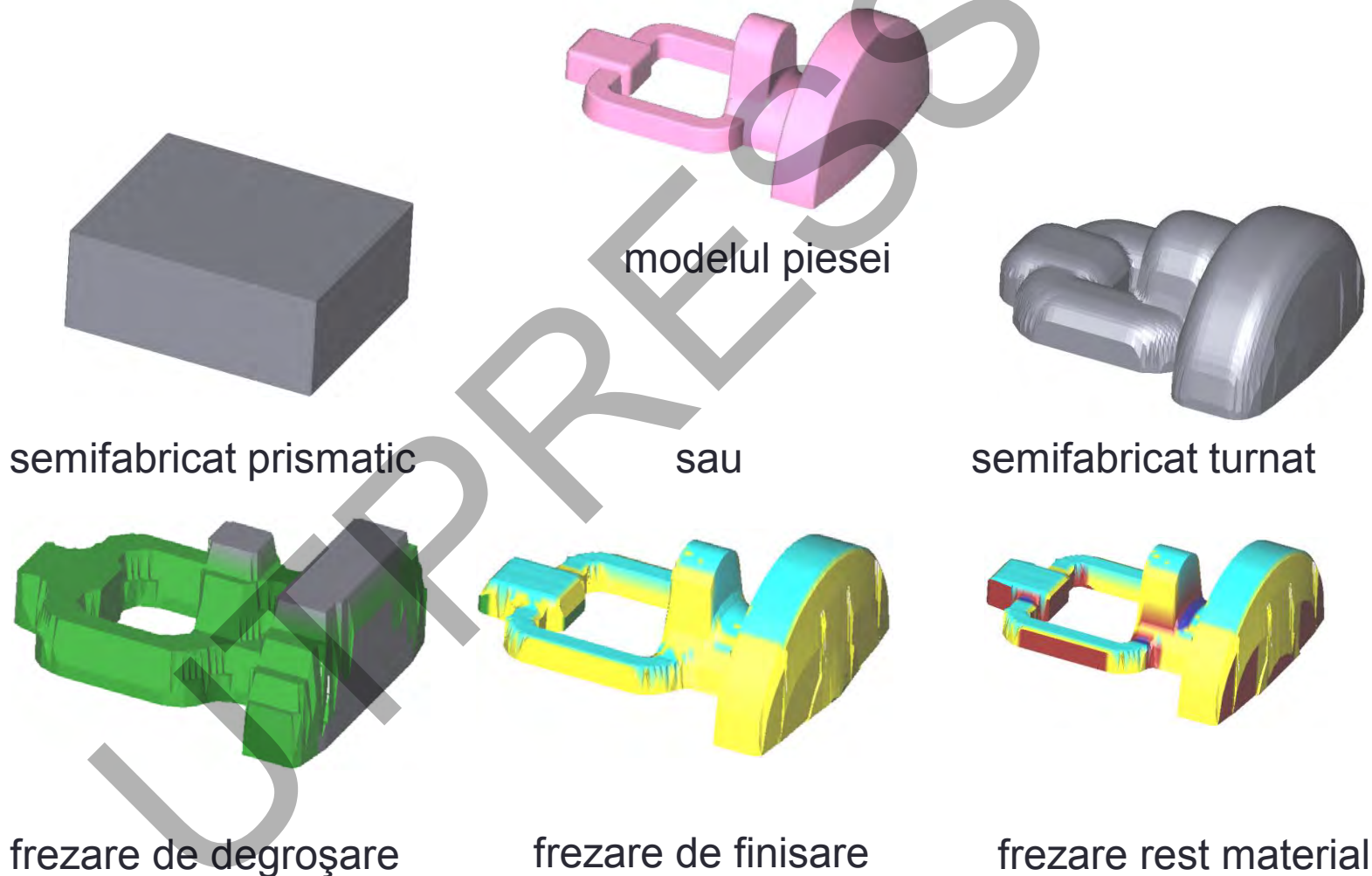
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

# Strategii de prelucrare

În SprutCAM se poate programa utilizarea următoarelor strategii de prelucrare:

degroșări și finisări de tip curbă de nivel (în plane orizontale succesive)

degroșări și finisări plane (în plane verticale succesive);

degroșări și finisări profilate după diverse traiectorii prestabilite (de-a lungul unor curbe pre-definite, etc. );

finisări combinate;

finisări complexe;

strategii de eliminare a restului de material rămas după finisări (restul de material este identificat automat);

prelucrarea suprafețelor plane orizontale;

prelucrarea pereților verticali drepecți;

prelucrarea teșiturilor;

eliminarea materialului din jurul unor arii definite (prelucrări de tip insulă și/sau buzunar);

prelucrări de gravare;

operații de găurire, alezare, filetare;

strunjiri de degroșare și finisare longitudinale și plane.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Strategii de degrosare

## Degroșare "Waterline"



Presupune îndepărtarea materialului semifabricatului aflat în afara modelului 3D al piesei. Materialul este îndepărtat prin deplasări ale sculei în plane orizontale (XY) aflate la cote Z diferite (curbe de nivel). Această strategie se recomandă, pentru o degroșare inițială, când diferențele de dimensiuni dintre semifabricat și modelul piesei sunt foarte mari.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

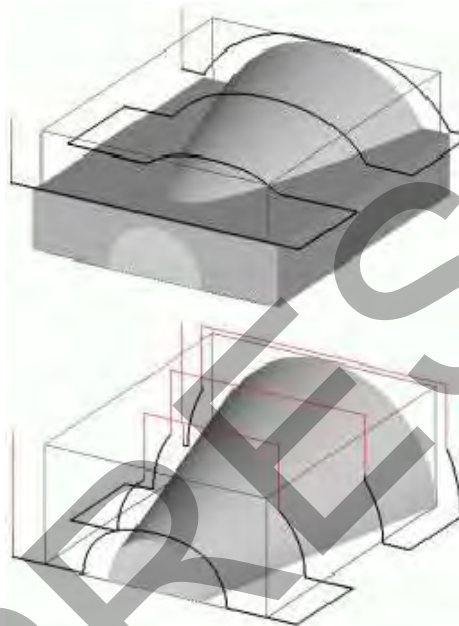
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## Degroșare “Plane”



Presupune îndepărtarea materialului semifabricatului aflat în afara modelului 3D al piesei. Secțiunile de lucru sunt amplasate în plane verticale paralele. Pentru a limita presiunea asupra sculei, prelucrarea poate fi realizată cu adâncimi Z pre-reglate de valori mici. Rezultatul prelucrării este de obicei mai apropiat de forma piesei finite decât rezultatul unei degroșări “Waterline” cu aceiași parametri. Această strategie se utilizează de obicei atunci când se dorește obținerea după degroșare a unui piese intermediare apropiate de forma piesei finite. De asemenea, este utilă la prelucrarea materialelor moi.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

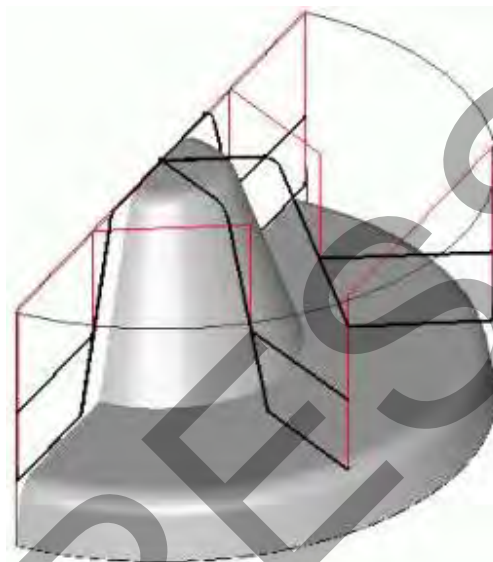
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## Degroșare “Drive”



Îndepărtarea materialului dintre semifabricat și model se face prin treceri separate.

În funcție de parametrii operație, trecerile sunt situate fie în plan vertical, fie în cilindrii verticali a căror formă și locație sunt definite de către curbele directoare de pe model.

Pentru a limita presiunea asupra sculei, prelucrarea poate fi realizată cu adâncimi Z pre-reglate de valori mici. Rezultatul prelucrării este uneori foarte apropiat de forma piesei finite, dar datorită valorii inegale a adaosului de prelucrare pe diferite treceri, nu este întotdeauna posibil să se obțină un timp de prelucrare optim. Operația dă cele mai bune rezultate numai pentru anumite forme de semifabricate și modele.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

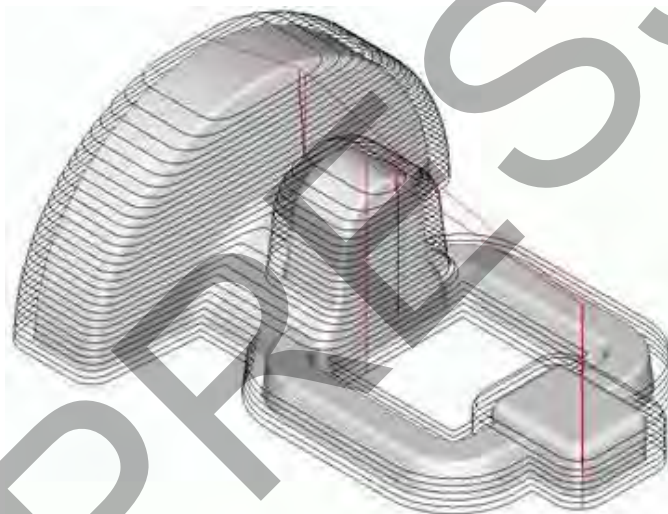
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Strategii de finisare

## Finisare "Waterline"



Prelucrarea tip curbă de nivel a suprafețelor unui model. Prelucrarea prin frezare este realizată prin deplasarea sculei în plane orizontale succesive. Operația dă rezultate bune când suprafețele principale ale modelului prelucrat sunt apropiate de verticală.

Pentru finisarea modelelor de mare complexitate este recomandat ca după finisarea "Waterline" să se utilizeze și alte strategii de finisare.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

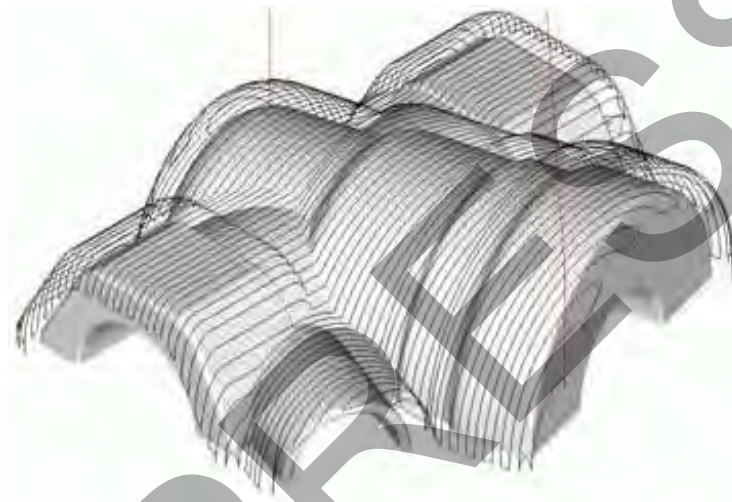
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## Finisare "Plane"



Traiectoriile sculei sunt plasate în plane verticale paralele. Rezultate bune se obțin atunci când suprafețele principale ale modelului prelucrat sunt apropiate de orizontală și/sau de verticală perpendicularare pe traiectoria sculei. Astfel, și în acest caz pentru prelucrarea modelelor complexe se recomandă utilizarea acestei operații împreună cu operația de finisare "Waterline" sau cu o altă finisare "Plane", cu traiectoriile sculei perpendicularare pe traiectoriile primei operații.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

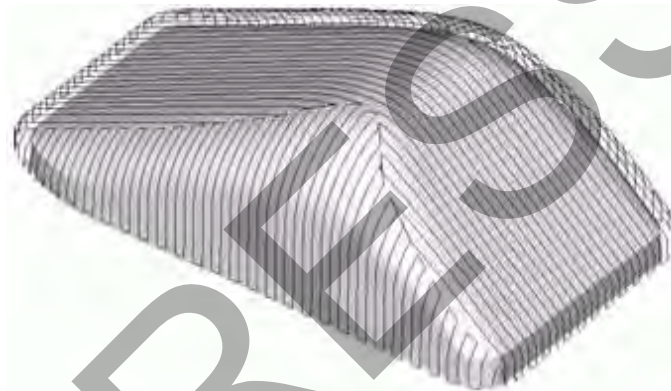
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## Finisare “Optimized Plane”



Ca rezultat al acestei operații se creează două operații de finisare “Plane” cu traiectoriile sculei respectiv perpendiculare unele pe celelalte. Parametrii implicați ai operației sunt stabiliți în așa fel încât fiecare din cele două operații componente vor asigura prelucrarea numai acelor suprafețe ale modelului pentru care se pot obține rezultate optime. Aceasta implică o calitate constantă a prelucrării pe întreaga suprafață a modelului. Utilizarea acestei operații permite o prelucrare corespunzătoare a modelelor cu suprafețe de formă complexă și permite minimizarea timpului de prelucrare.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

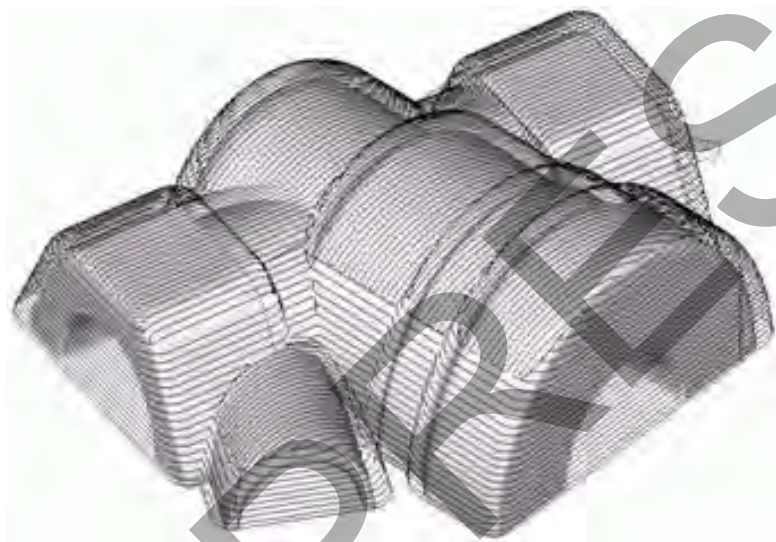
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## Finisare “Complex”



Prin această strategie de prelucrare sunt generate automat două operații: finisare “Plane” și finisare “Waterline” pentru modelul 3D al piesei. Parametrii impliciți ai celor două operații sunt stabiliți automat astfel încât suprafețele apropiate de orizontală sunt prelucrate utilizând operația de finisare “Plane”, iar suprafețele apropiate de verticală utilizând operația de finisare “Waterline”.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

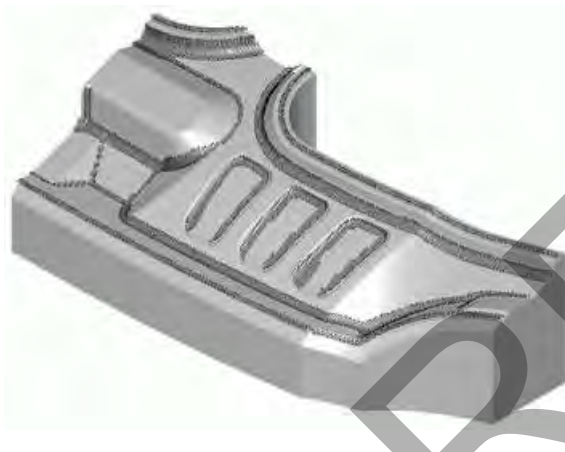
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## Îndepărtare rest material "Plane"



În cazul prelucrărilor de tip îndepărtare rest material, materialul rămas după operațiile de degroșare și finisare este identificat automat. Operația presupune îndepărtarea restului de material utilizând o strategie de prelucrare de tip finisare "Plane". Suprafețele insuficient prelucrate după operațiile de degroșare și finisare anterioare sunt frezate utilizând treceri situate în plane verticale paralele. Operația este utilizată la re-prelucrarea suprafețelor ușor înclinate (aproape de orizontală) și a suprafețelor apropiate de verticală perpendiculare (sau aproape perpendiculare) pe traiectoriile sculei.

# Scule utilizate

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

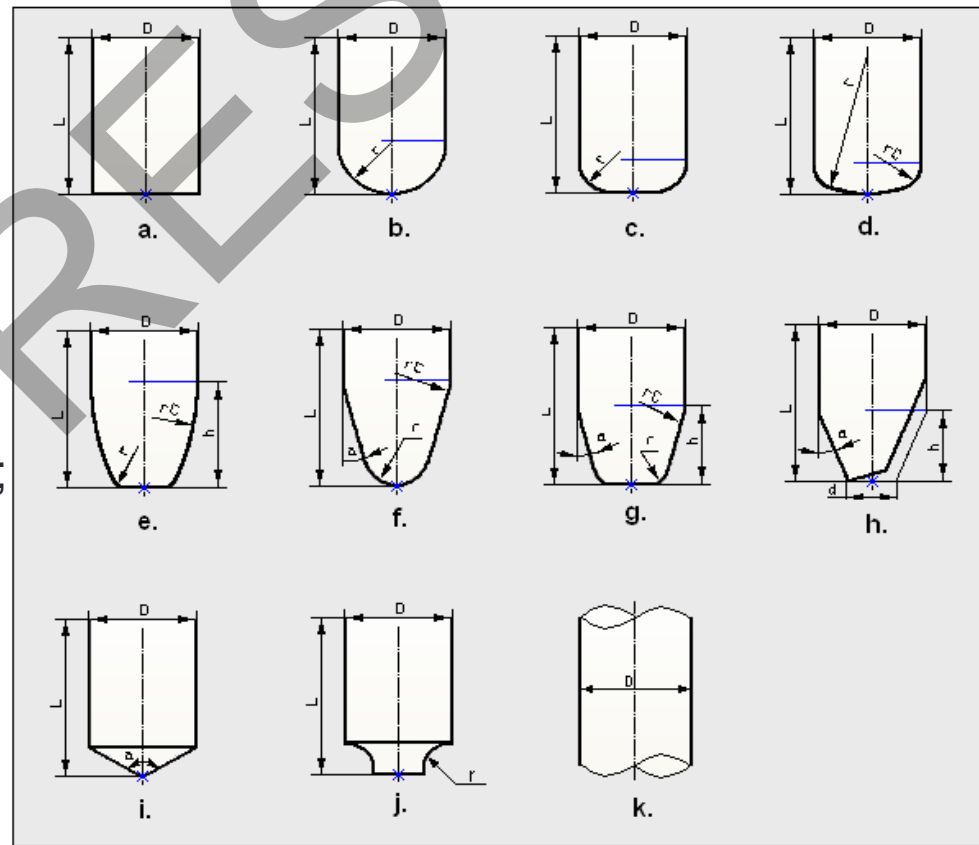
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Principalele tipuri de scule utilizate în SprutCAM sunt:

freze cilindro-frontale;  
freze cu cap sferic;  
freze toroidale;  
freze dublu radiale;  
freze dublu radiale limitate;  
freze conice;  
freze conice limitate;  
freze pentru gravare;  
burghie;  
freze cu rază la vârf negativă;  
scule pentru debitare;



# Prezentare succintă

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

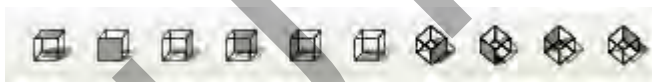
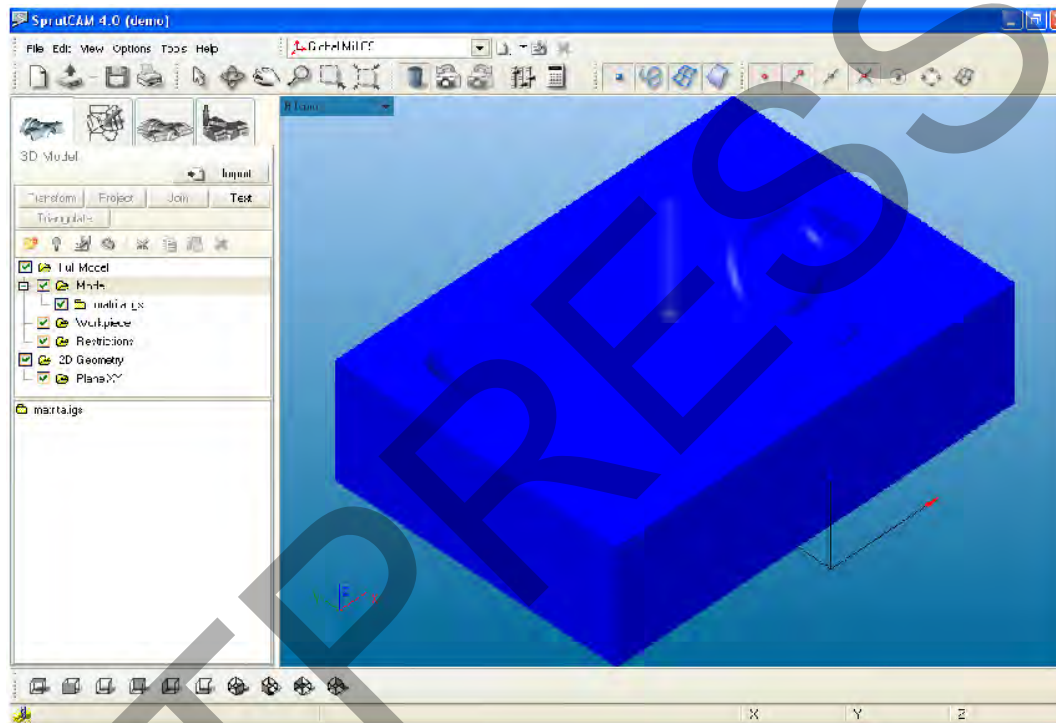
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM



**Butoane selectare mod  
vizualizare**



**Butoane selecție  
panoramare, focalizare**

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

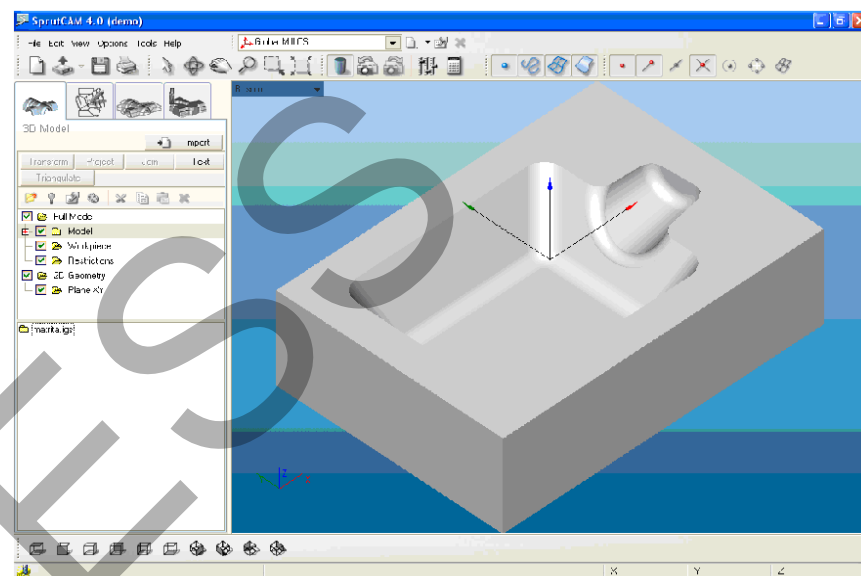
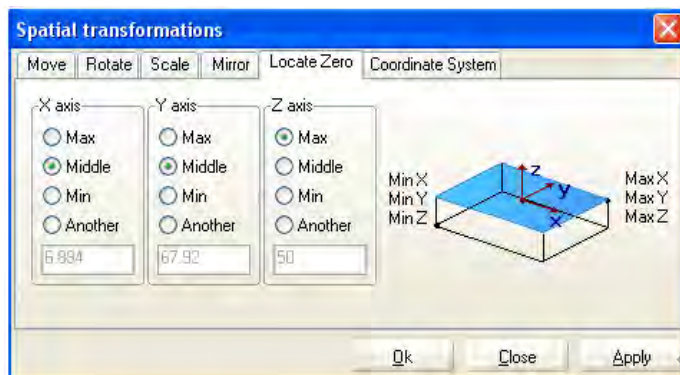
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

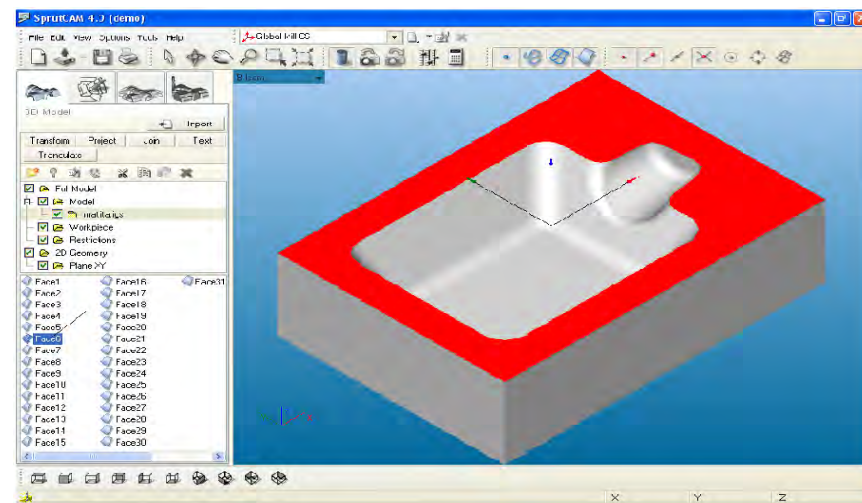
Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM



Intervenții asupra modelului  
geometric:  
Selectarea zonei ce urmează a fi  
prelucrată  
Soluție-curbe de restricție



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

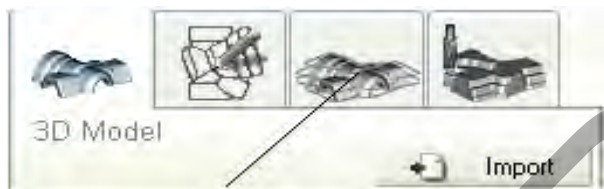
Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

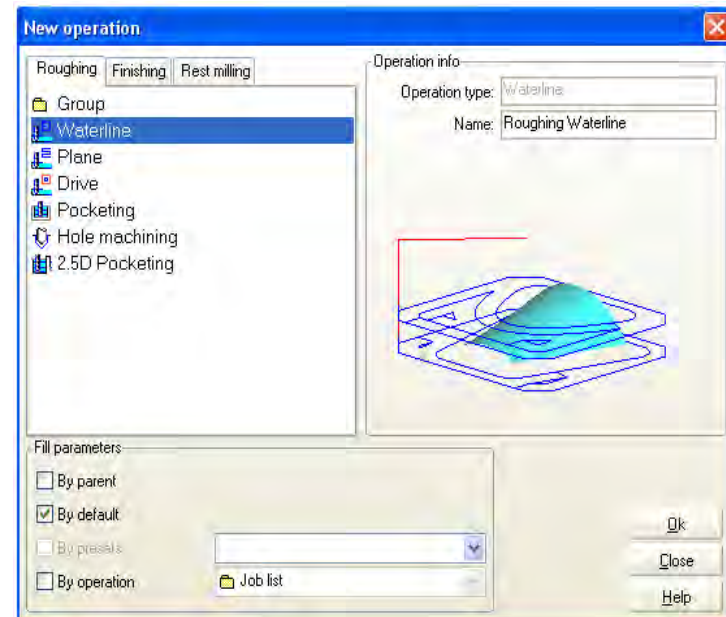
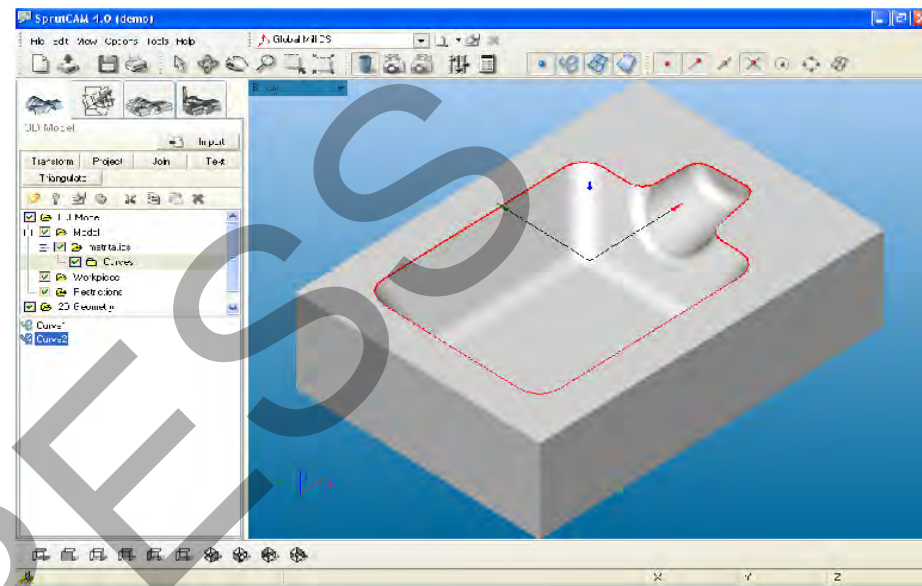
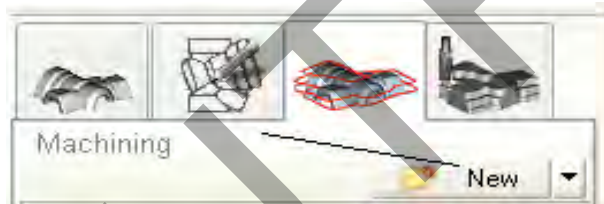
Cap.9.CAM

# Prelucrare

Selectare curba interioară a  
suprafeței (roșii)



Machining





Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

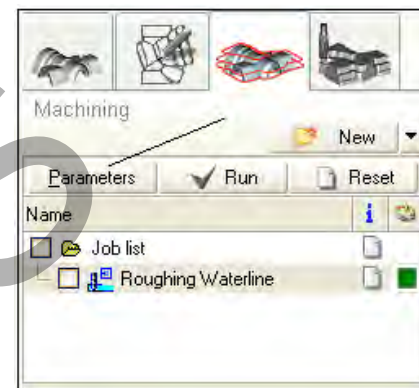
Cap.6. Cicluri  
fixe

Cap.7.  
Subprogram

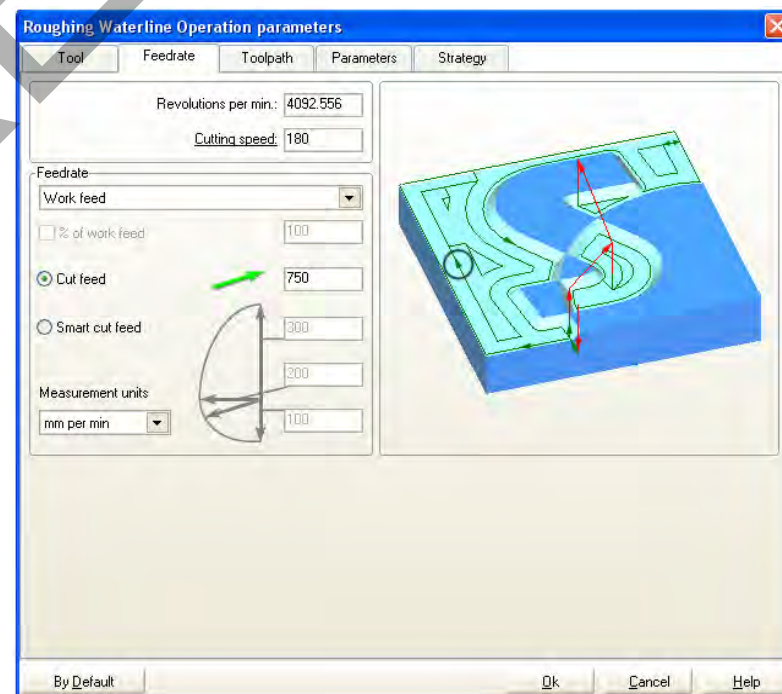
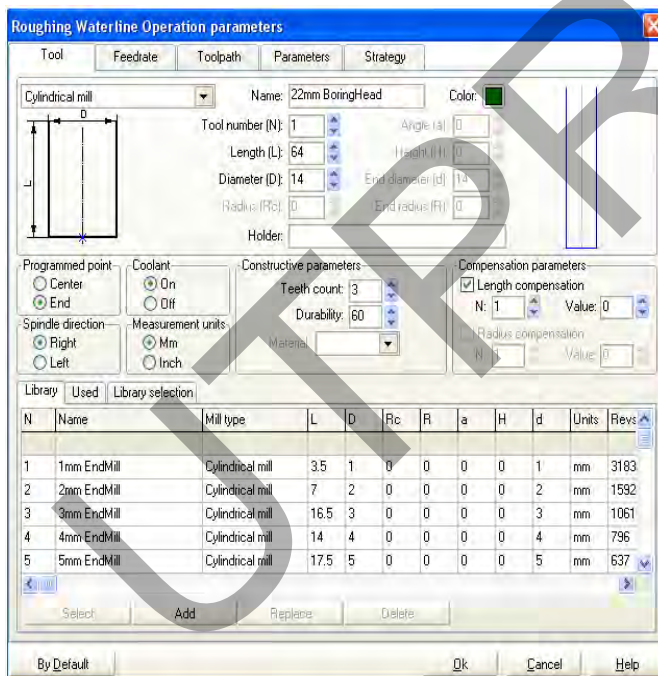
Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

Modificarea parametrilor operației selectate  
Parametrii sculei  
Parametrii de aşchiere  
Parametrii (Toolpath)



Selectarea automată poate fi actualizată de utilizator



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

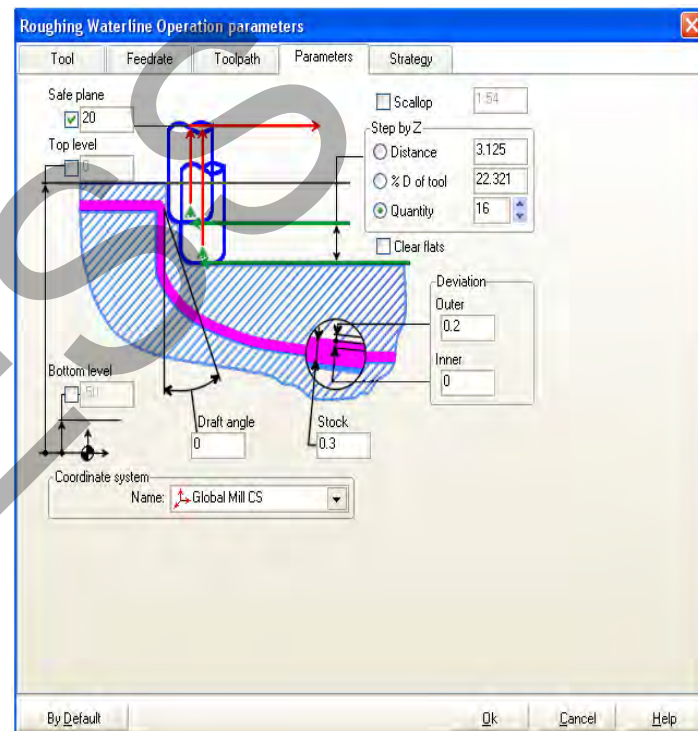
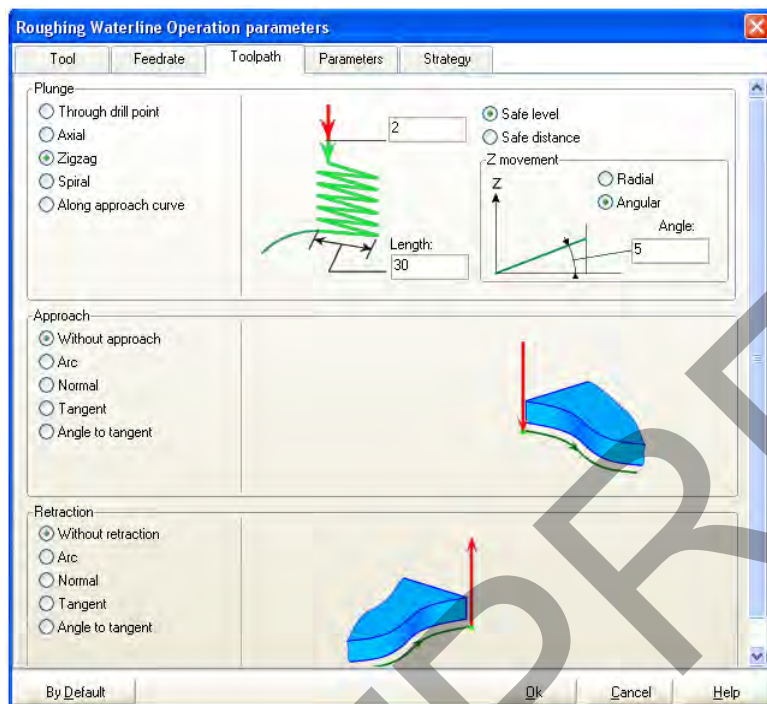
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Opțiuni pentru:

Alegerea modului de pătrundere  
Alegerea curbei de apropiere  
Alegerea curbei de îndepărtare

Alegerea valorii pentru "Safe level, Safe plane"  
Alegerea valorii pentru "Draft angle"  
Alegerea valorii pentru adaosul de finisare

# Traectoria sculei după operația de degrosare

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limba ISO

Cap.4.  
Programarea  
deplasărilor

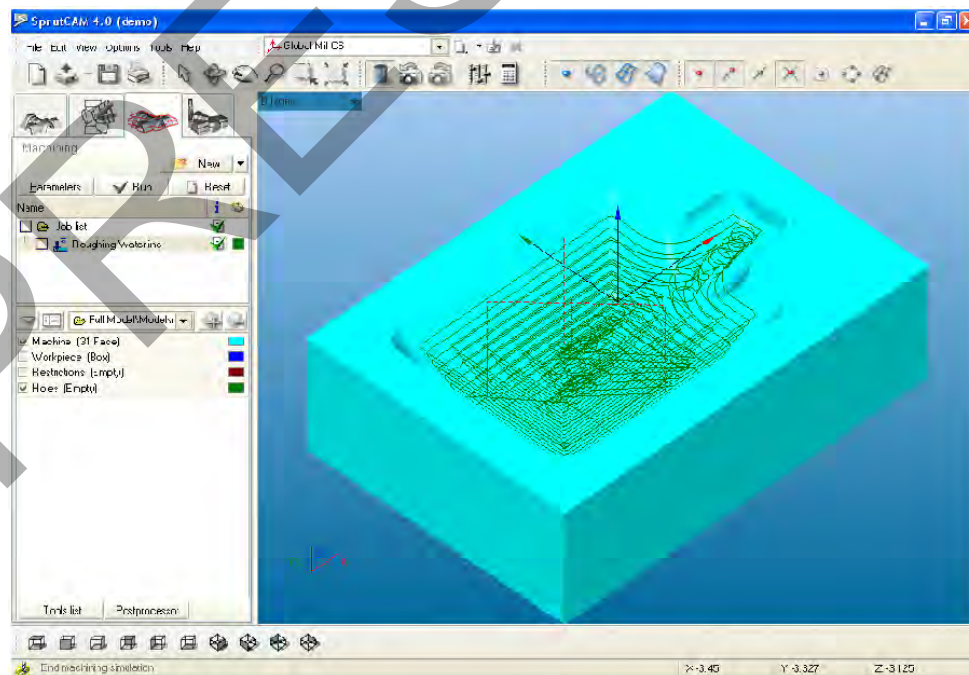
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Finisare

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

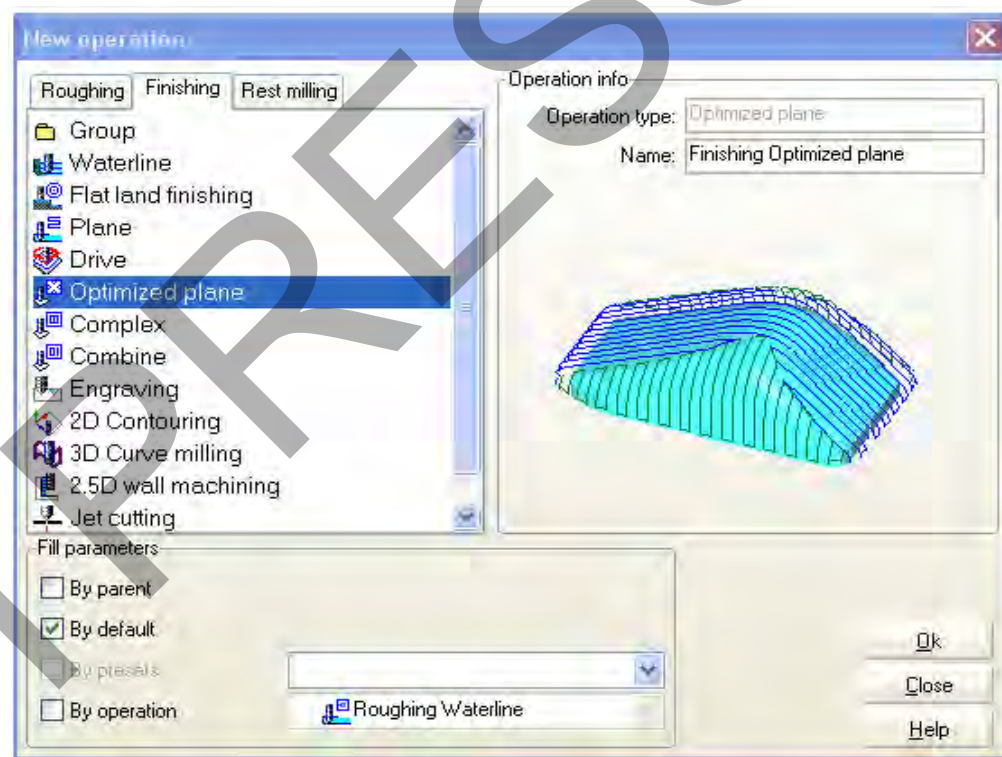
Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

Butonul "New"

Apoi se alege strategia „Optimized Plane”



Definirea sculelor – vezi operația de degrosare

# Definirea parametrilor „Strategies”

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

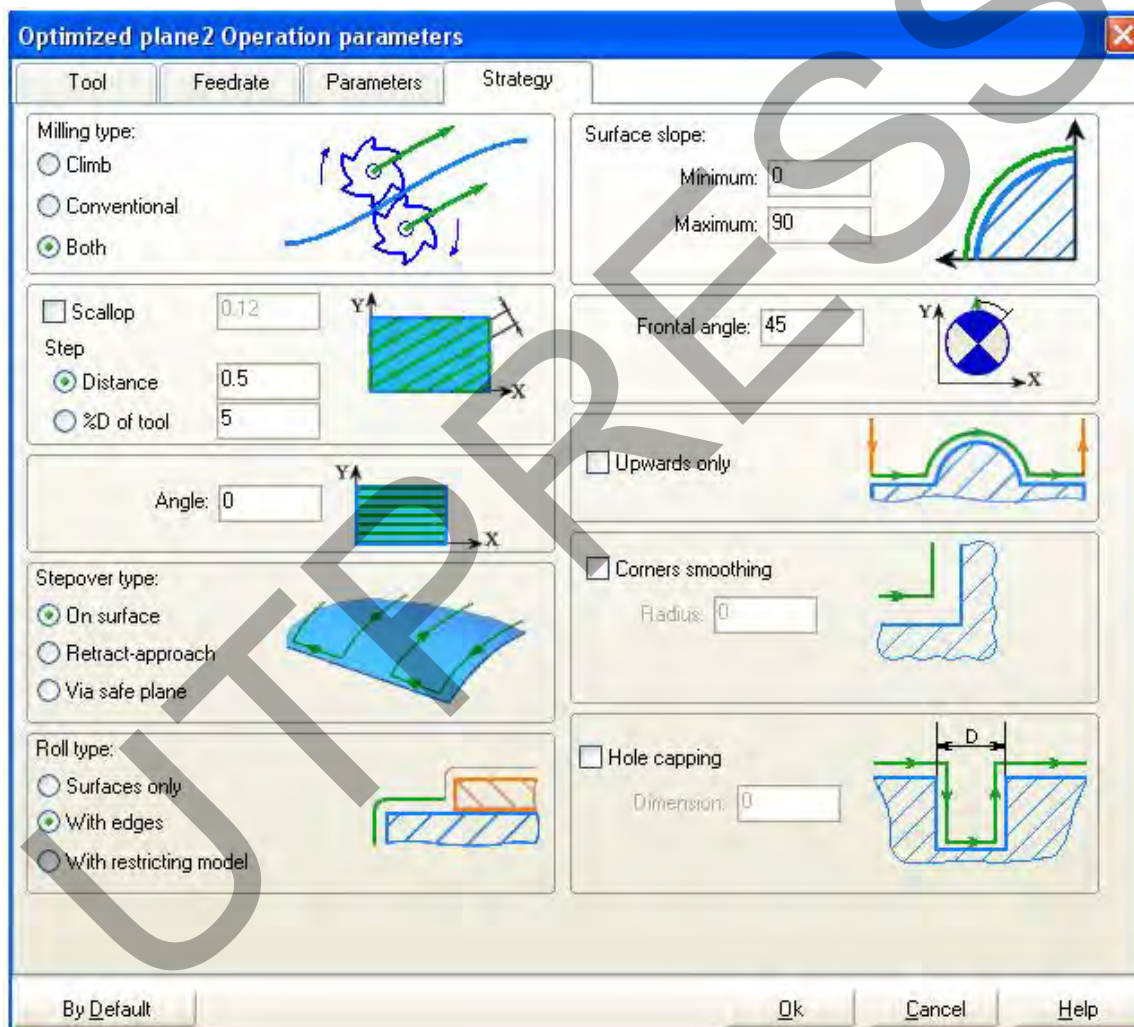
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

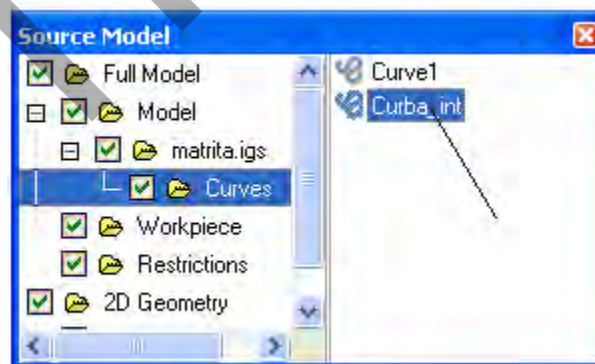
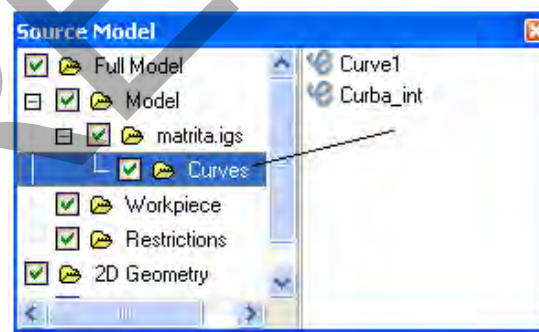
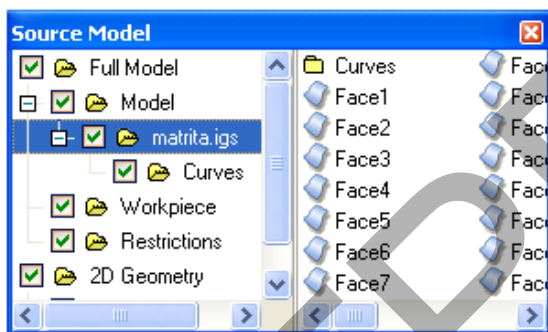
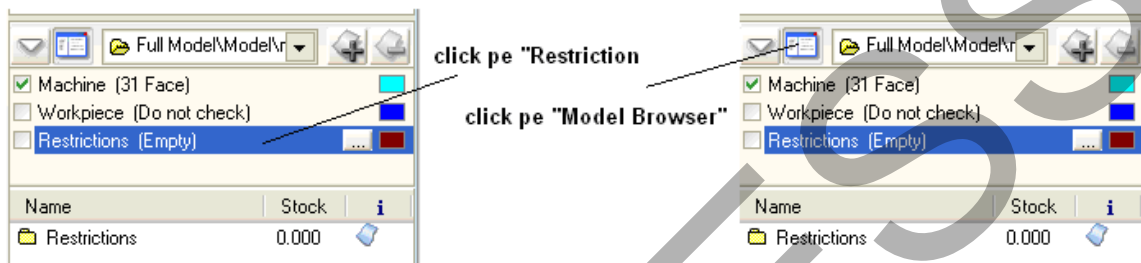
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Curbe de limitare: instrumente utile în definirea operațiilor de prelucrare  
Delimitează o zonă care nu/se dorește a fi prelucrată.



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

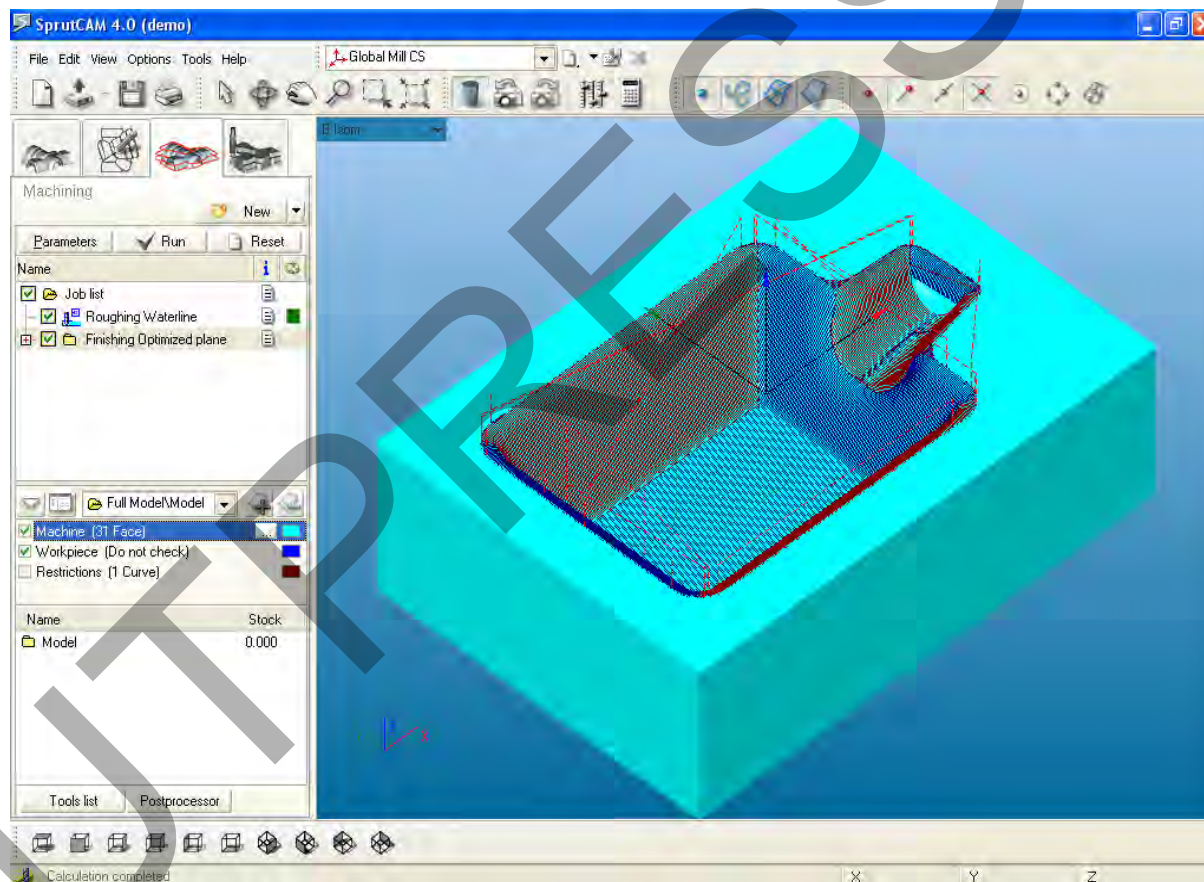
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

## Traectoria sculei la operația de finisare



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

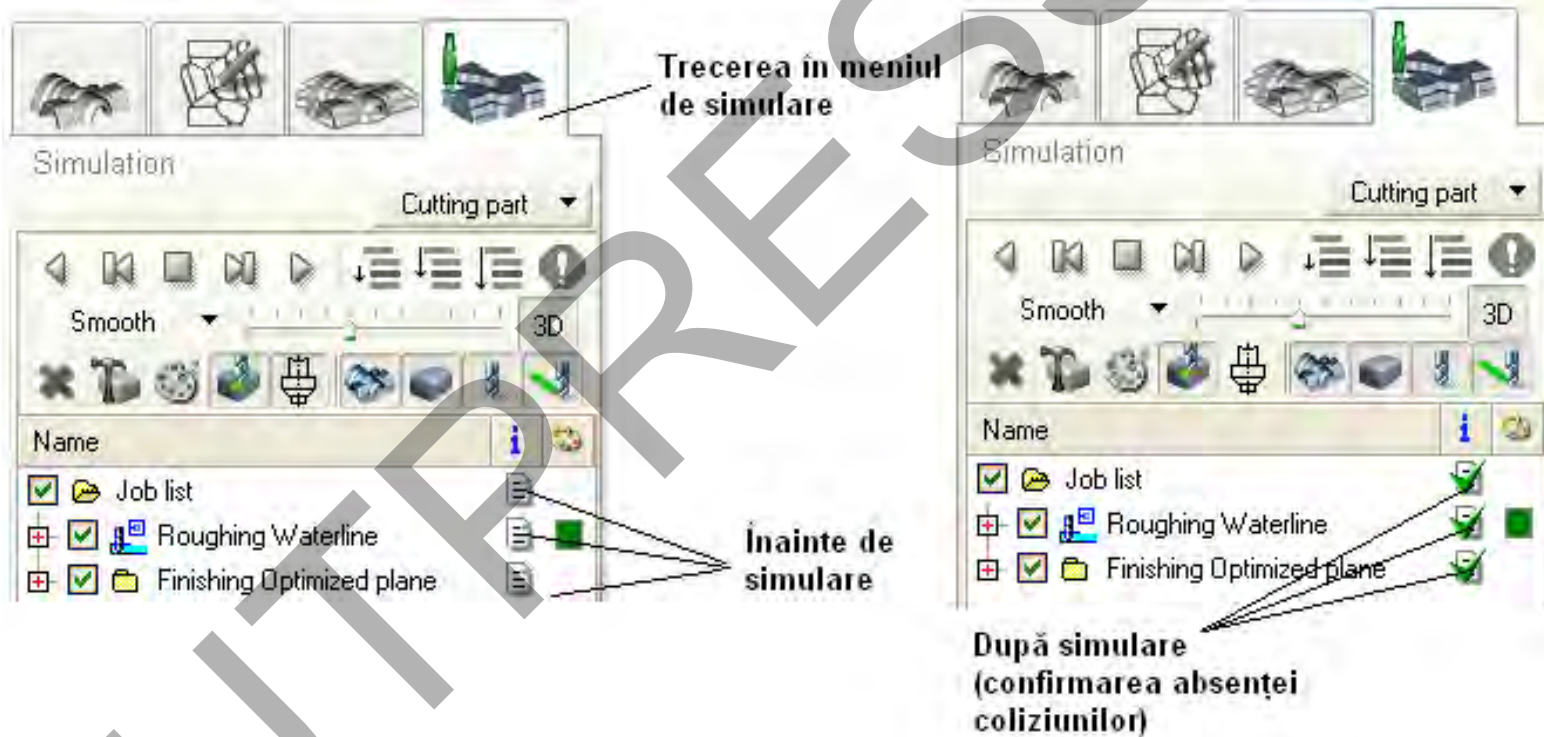
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

## Meniul de simulare





Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

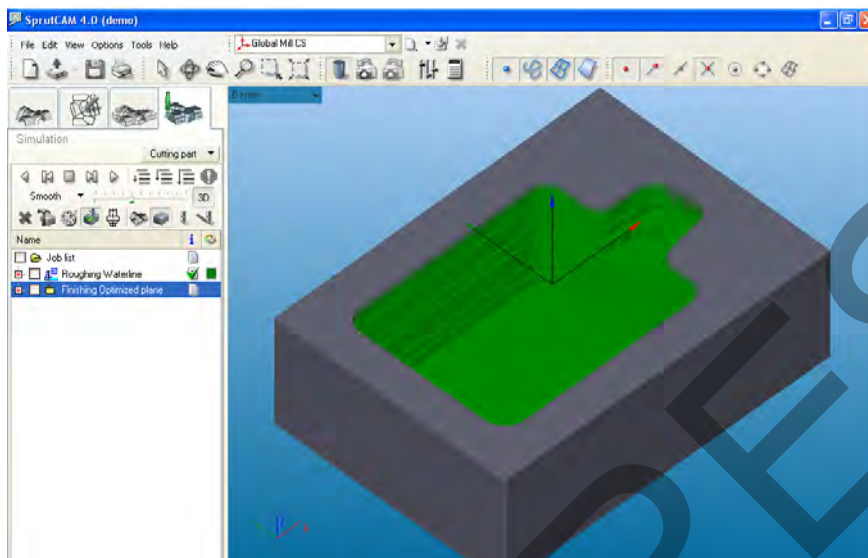
Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

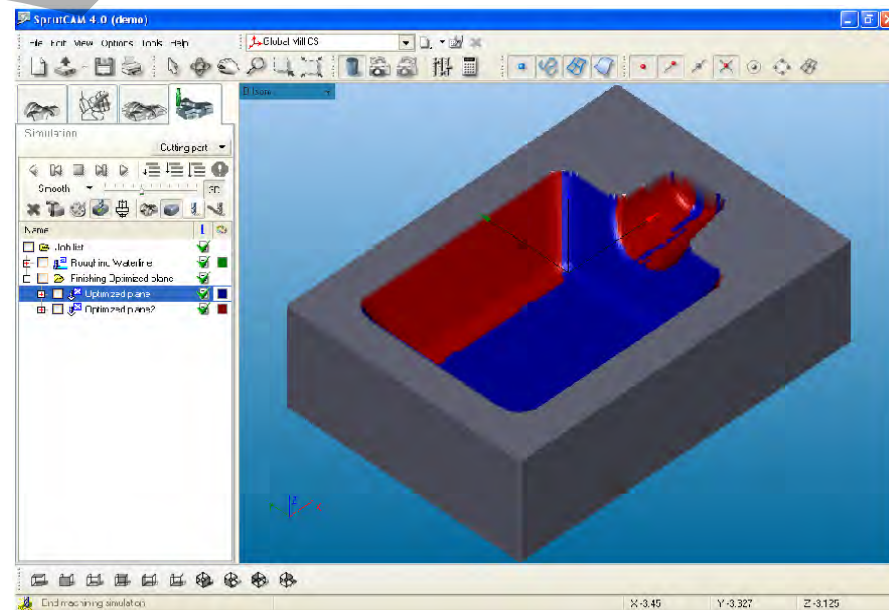
Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Piesa după simularea operației  
de degroșare

Piesa după simularea operației  
de finisare



# Frezare 2.5 D: modelul 3D al piesei și semifabricatul propus

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

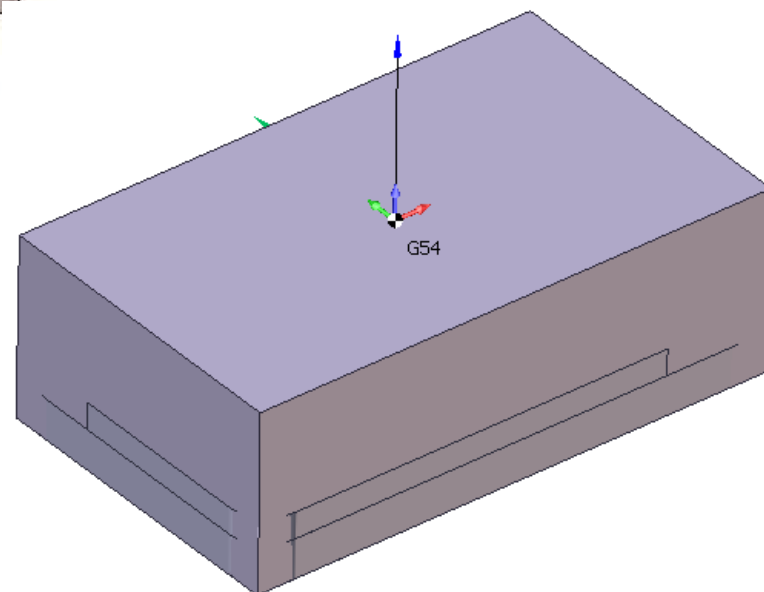
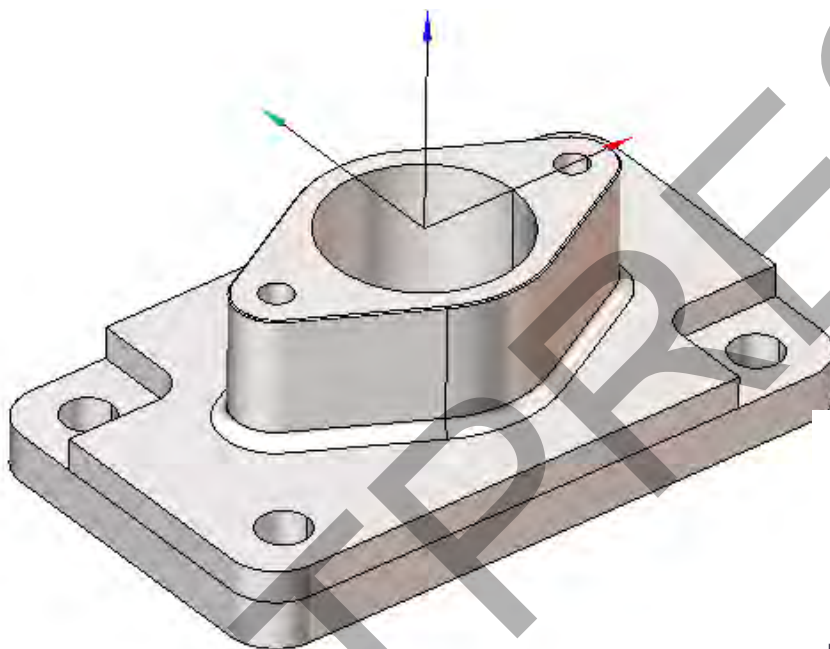
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

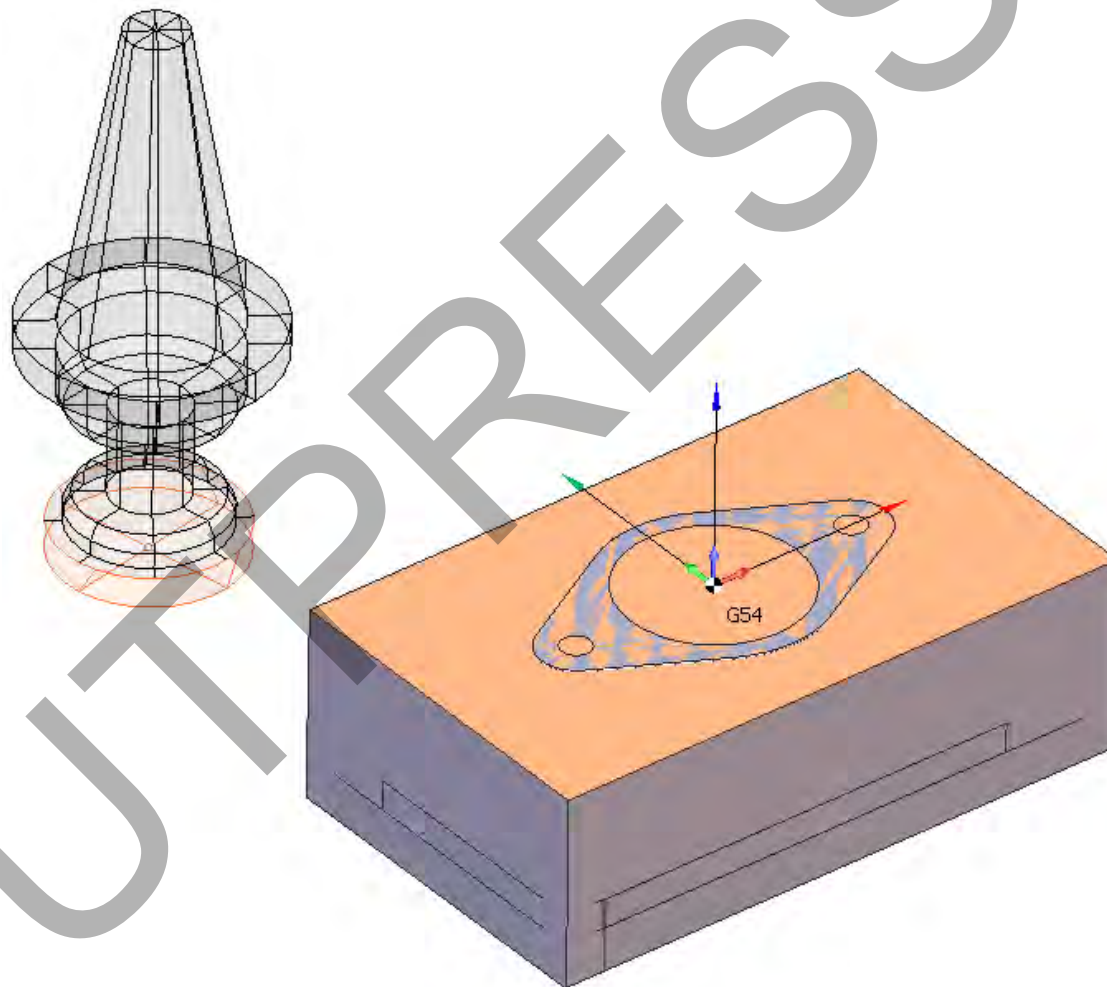
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Frezare 2.5 D: degroșare plană



# Frezare 2.5 D: degroșare de tip curbă de nivel

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

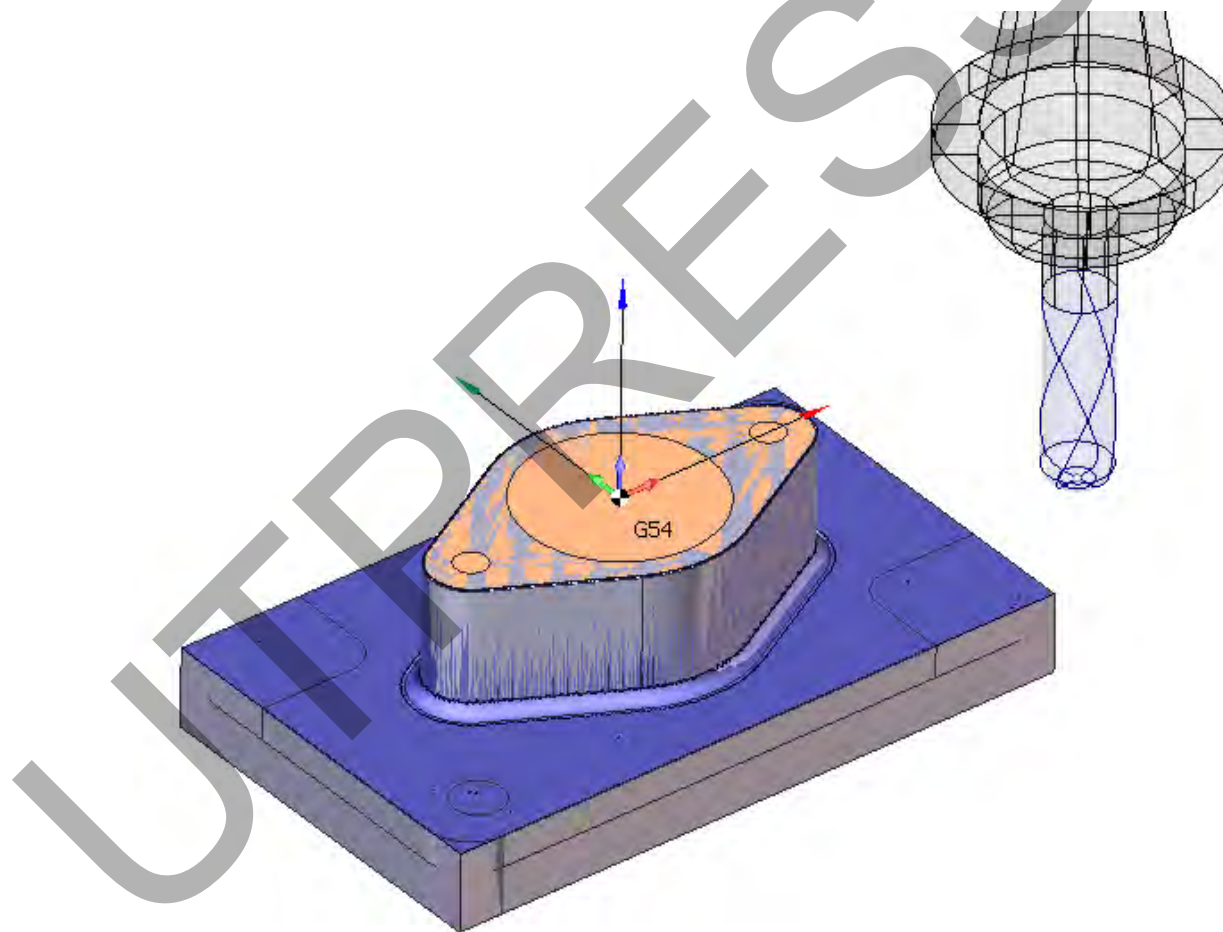
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Frezare 2.5 D: frezare gaură centrală

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

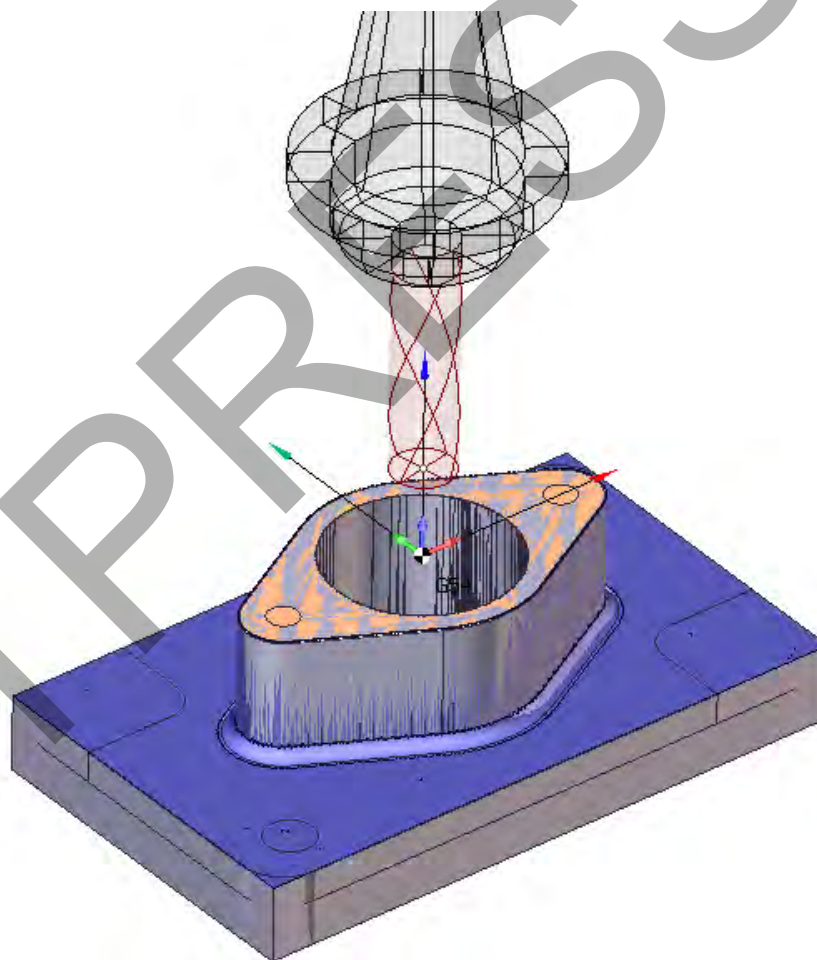
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

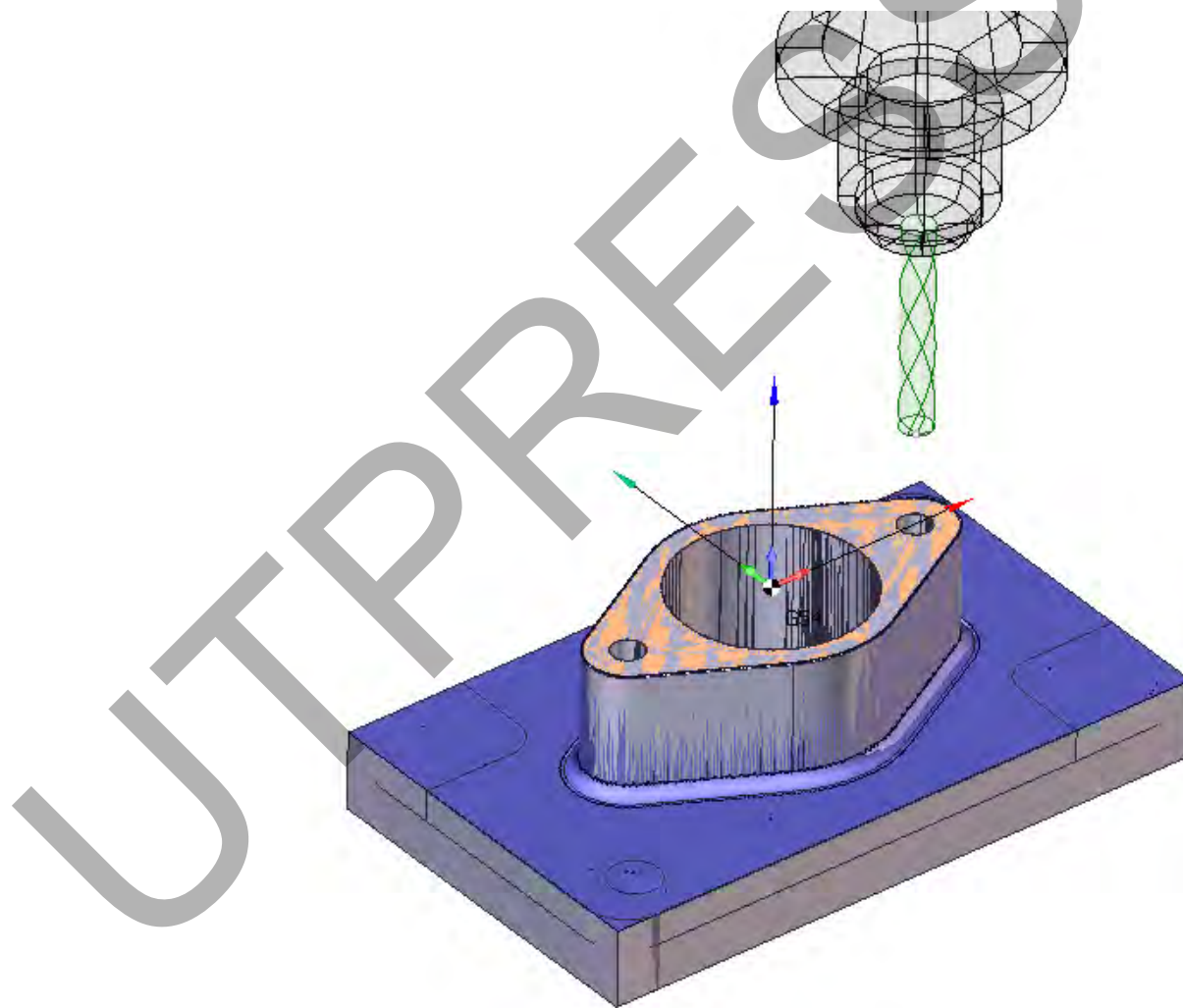
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Frezare 2.5 D: găurire găuri $\varnothing 10$



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

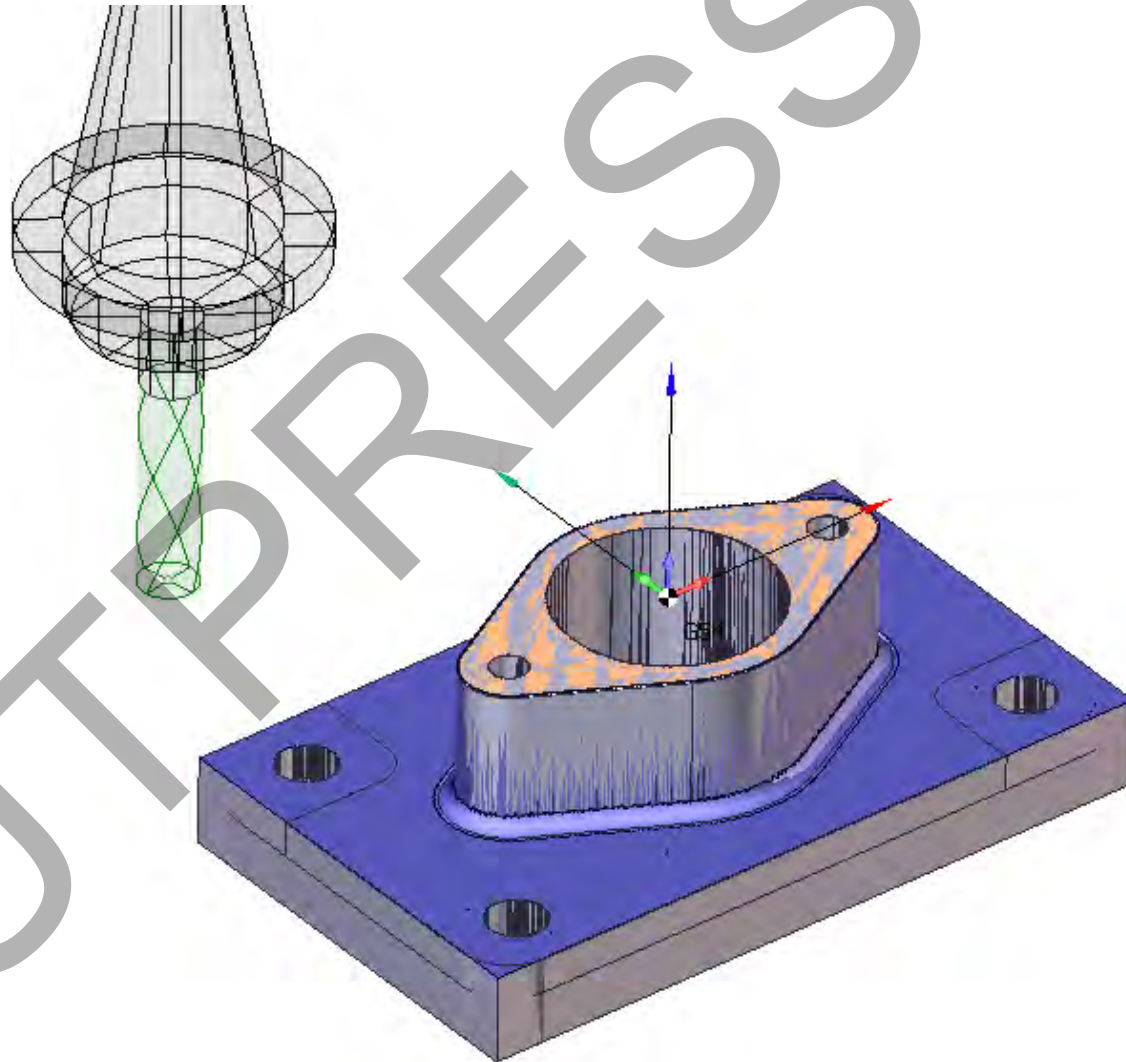
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

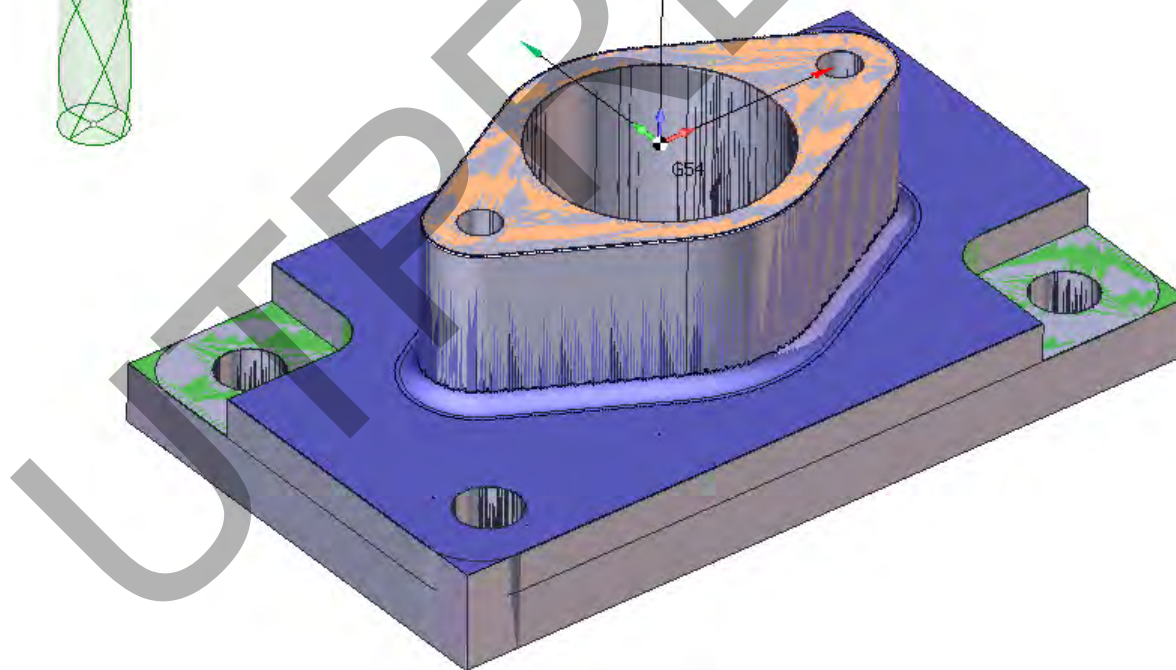
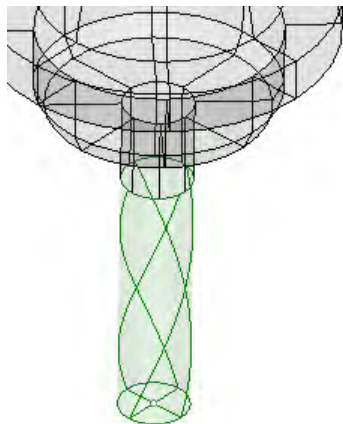
Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Frezare 2.5 D: găurire găuri $\varnothing 16$



# Frezare 2.5 D: frezare plană de finisare



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Frezare 2.5 D: frezare finisare colțuri

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

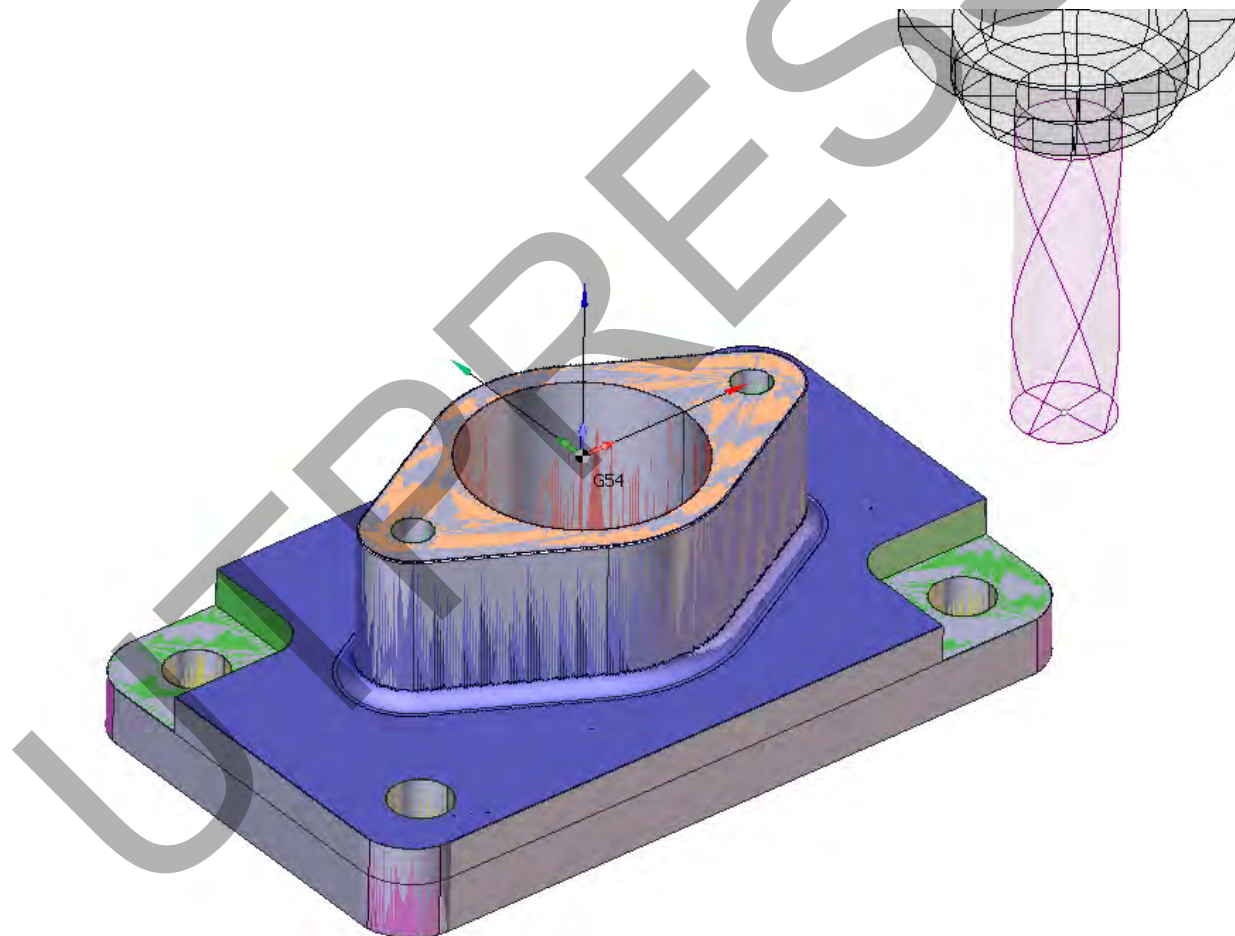
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

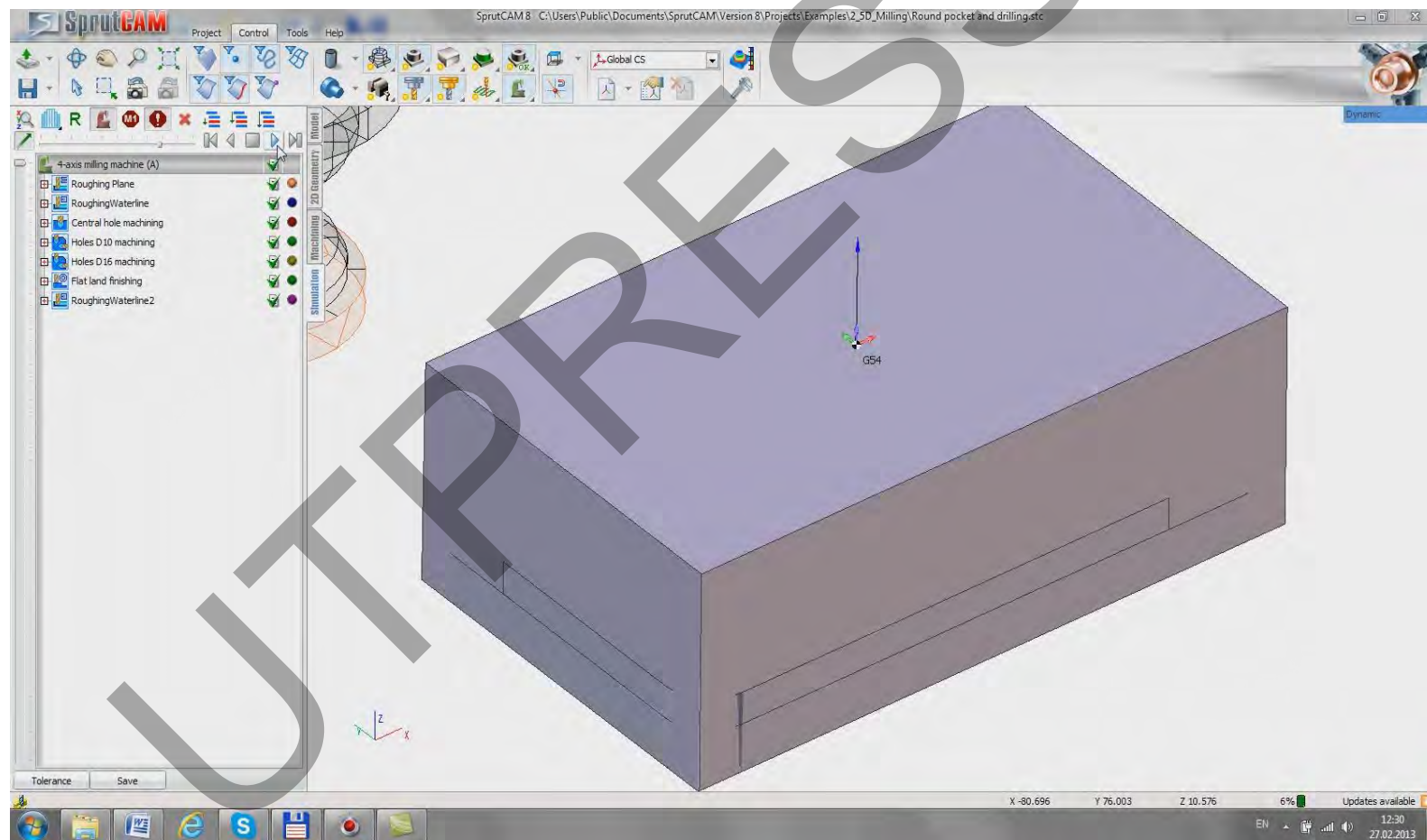
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

# Frezare 2.5 D: simulare



## Frezare 2.5 D: fragment din codul NC generat

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

```

.....
G03X-24.113Y40.382I-21.358J-41.906F100
G01X-62.137Y19.535
G03X-60.282Y-20.592I12.16J-19.544
G01X-22.151Y-41.445
G03X24.518Y-40.16I22.209J41.545
G01X62.137Y-19.535
G03X60.282Y20.592I-12.143J19.545
G01X21.305Y41.873
G00Z20.
X94.468Y73.119
Z-28.243
G01Z-29.243F50
G03X96.56Y69.914I4.914J0.923F100
G02X109.92Y58.858I-44.574J-67.458F200
G01Y43.434F600
G03X84.724Y65.354I-59.928J-43.445F200
G01X76.371Y69.914
X55.505F600
X82.961Y54.886F200
G02X109.92Y22.508I-33.041J-54.922
G03X112.518Y19.698I4.69J1.731
G00Z20.
.....

```

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

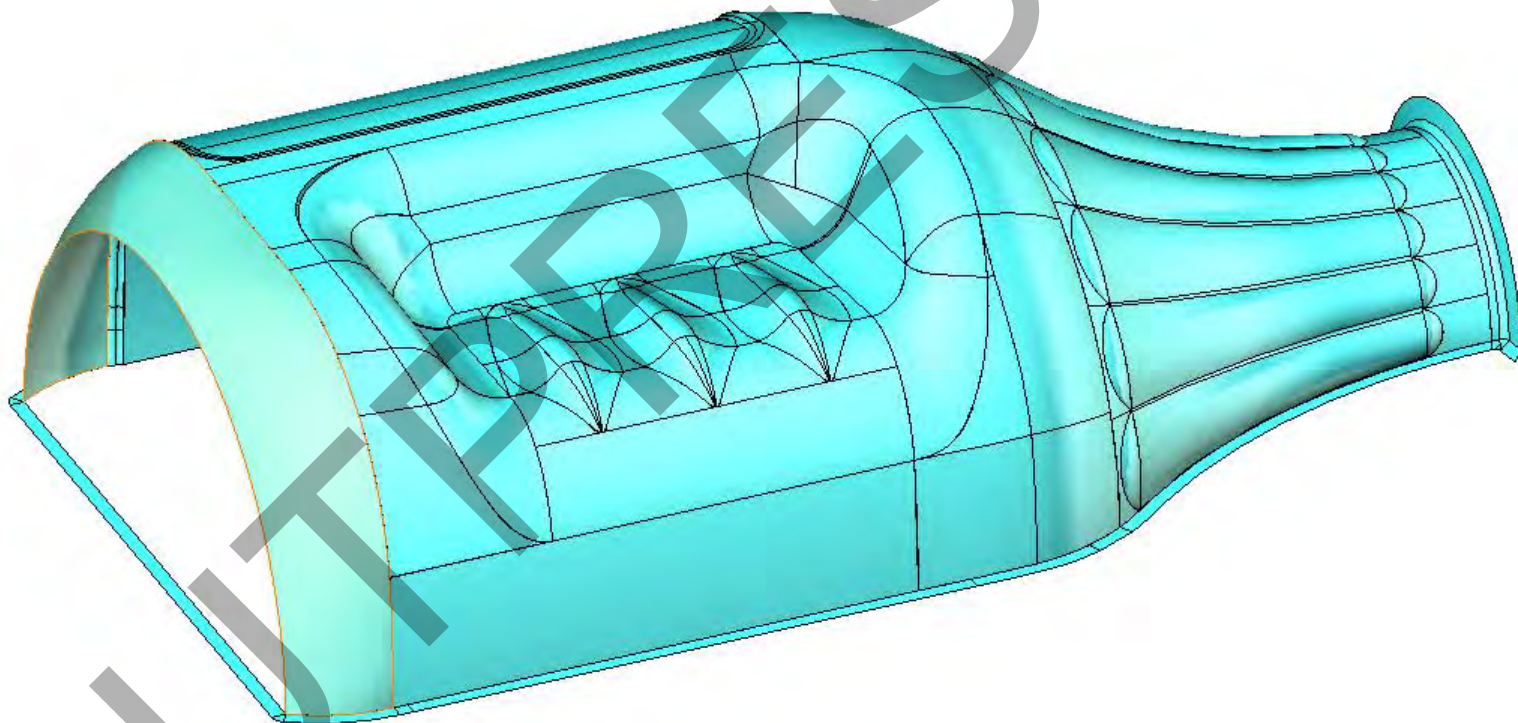
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Frezare 3D: modelul 3D al piesei



# Frezare 3D: semifabricat

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

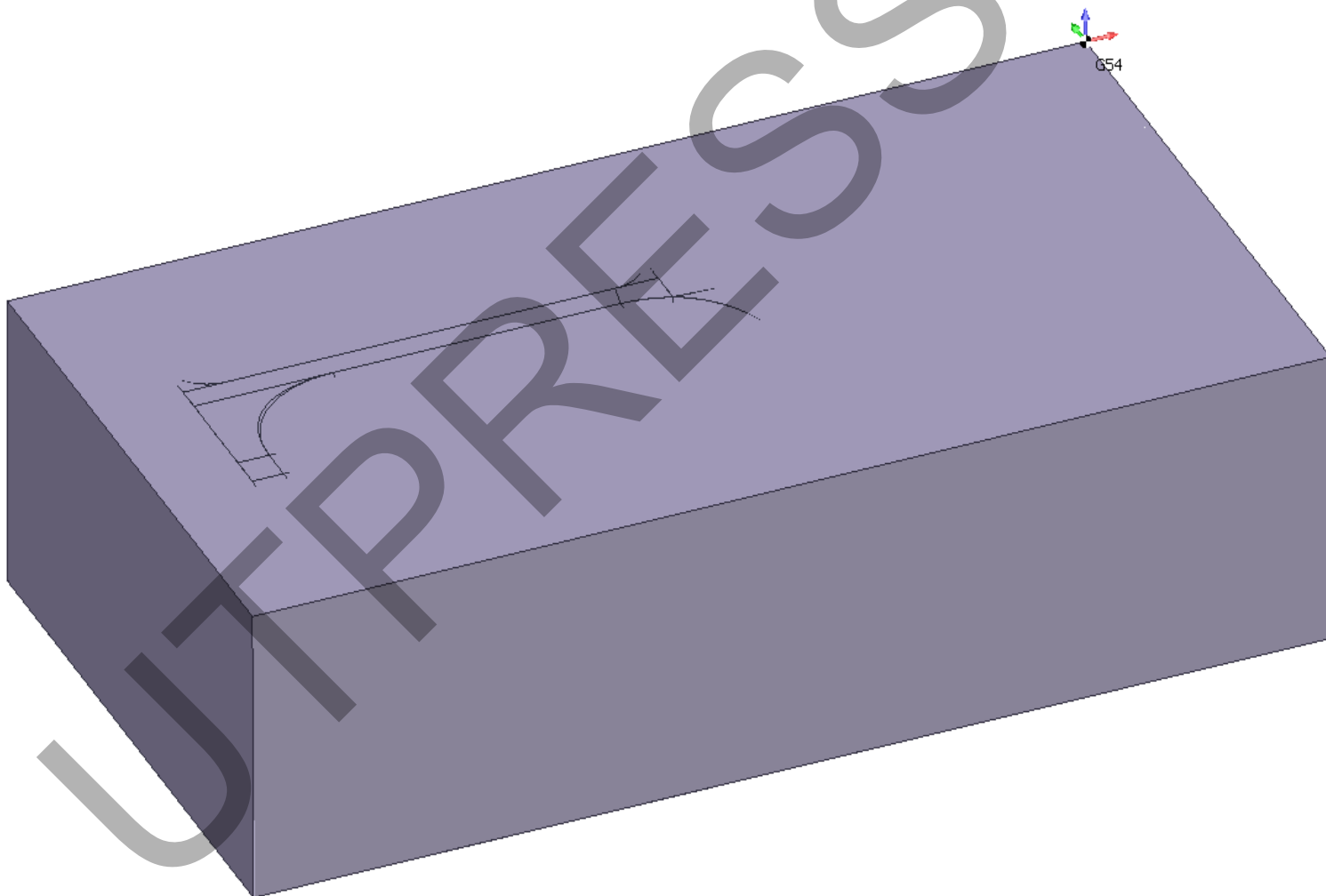
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Frezare 3D: degroșare de tip curbă de nivel

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

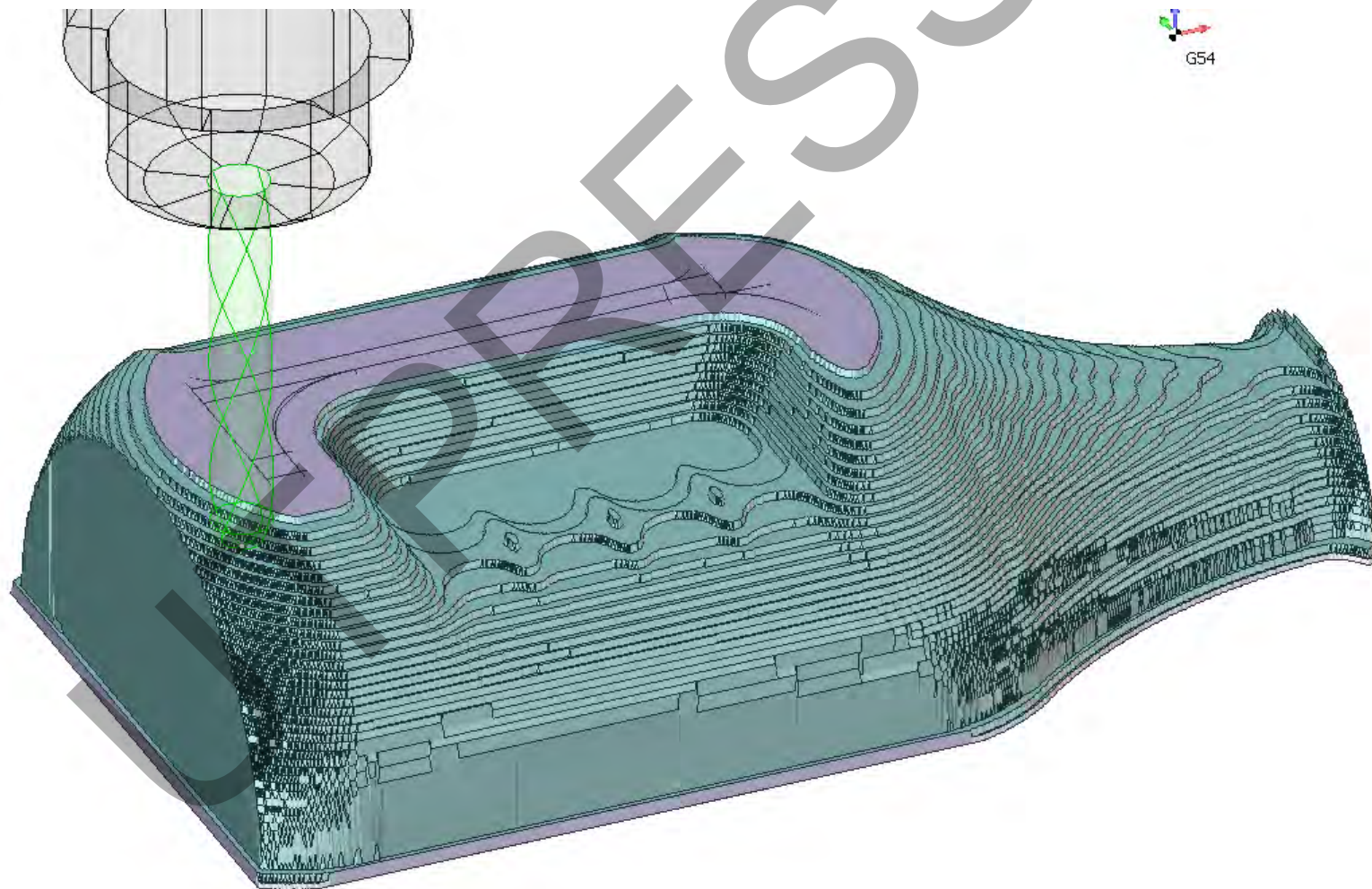
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Frezare 3D: finisare plană (aspect din timpul prelucrării)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

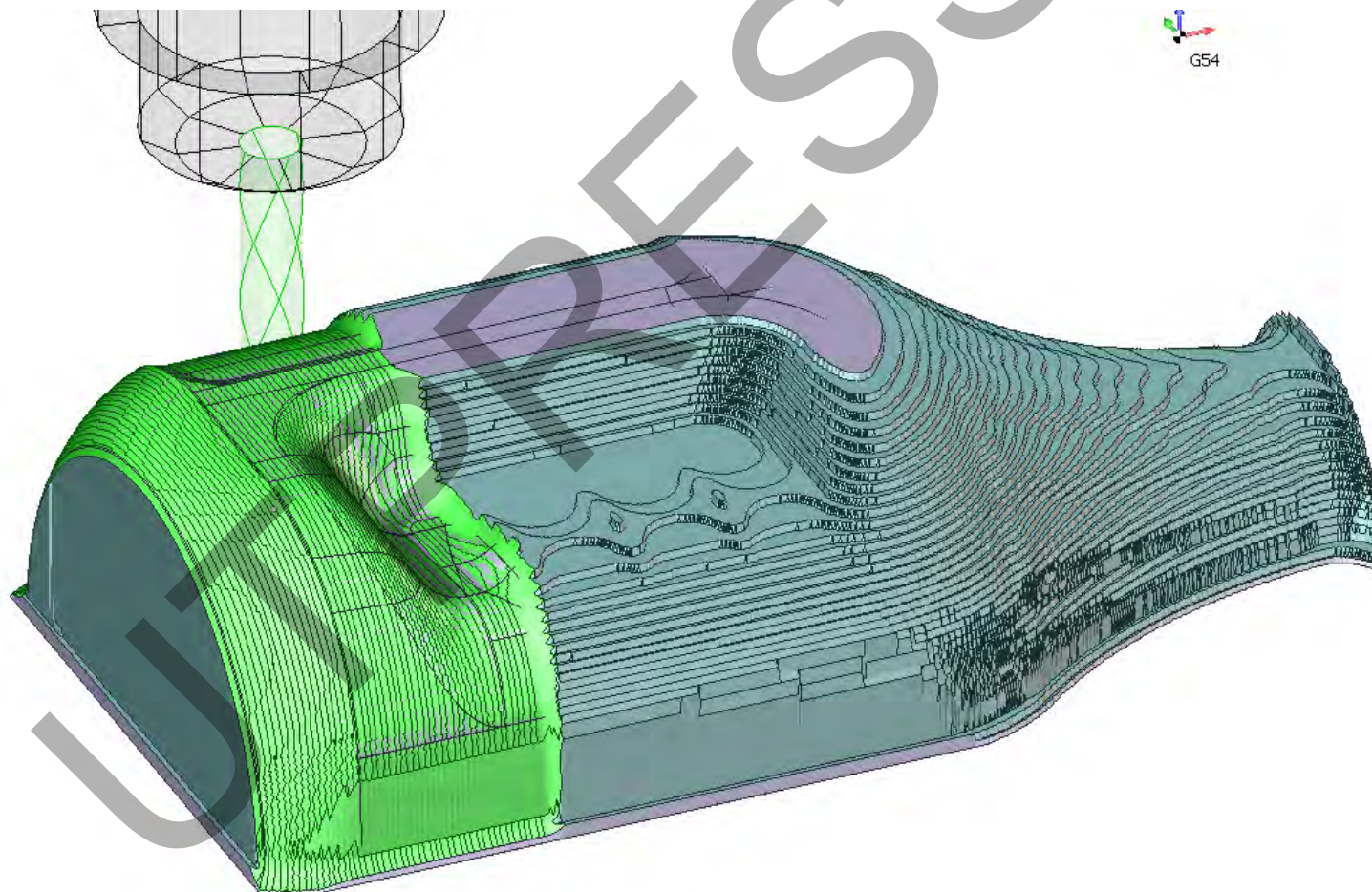
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

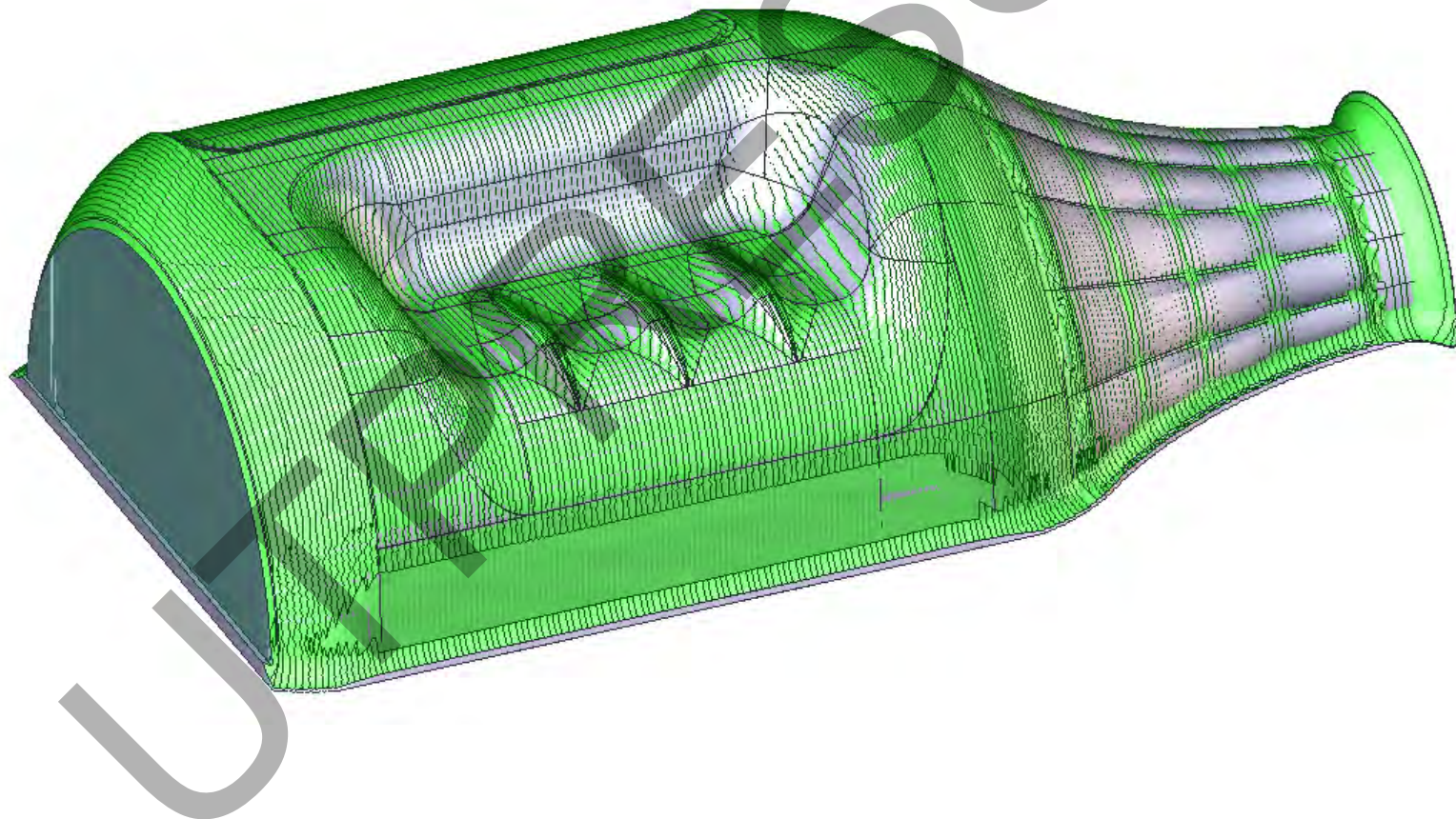
Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

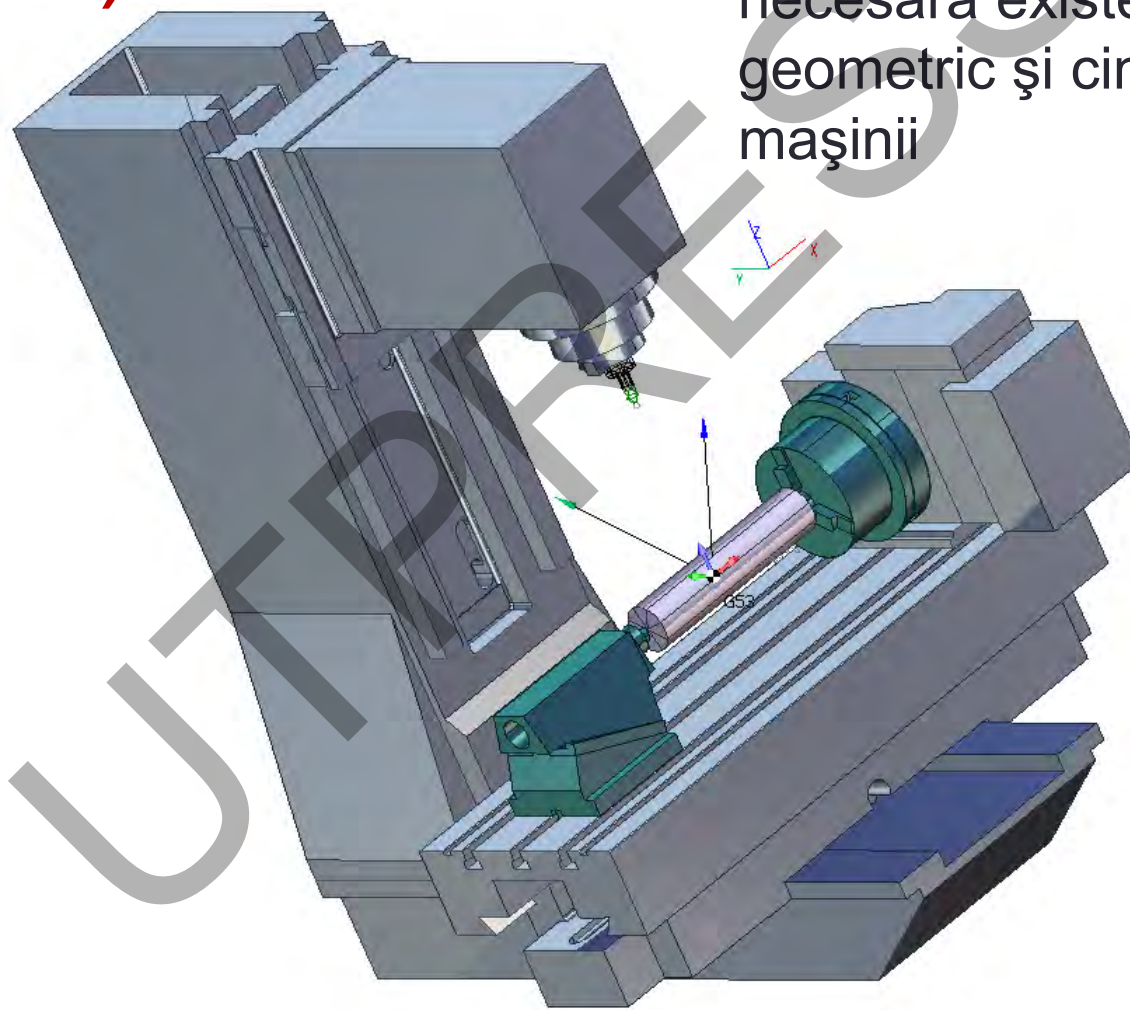
# Frezare 3D: finisare plană





# Frezare continuă în 4 axe (X, Y, Z, A)

Complexitate operațiilor face necesară existența modelului geometric și cinematic al mașinii



Se simulează și deplasările elementelor mobile ale mașinii

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Frezare continuă în 4 axe (X, Y, Z, A)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

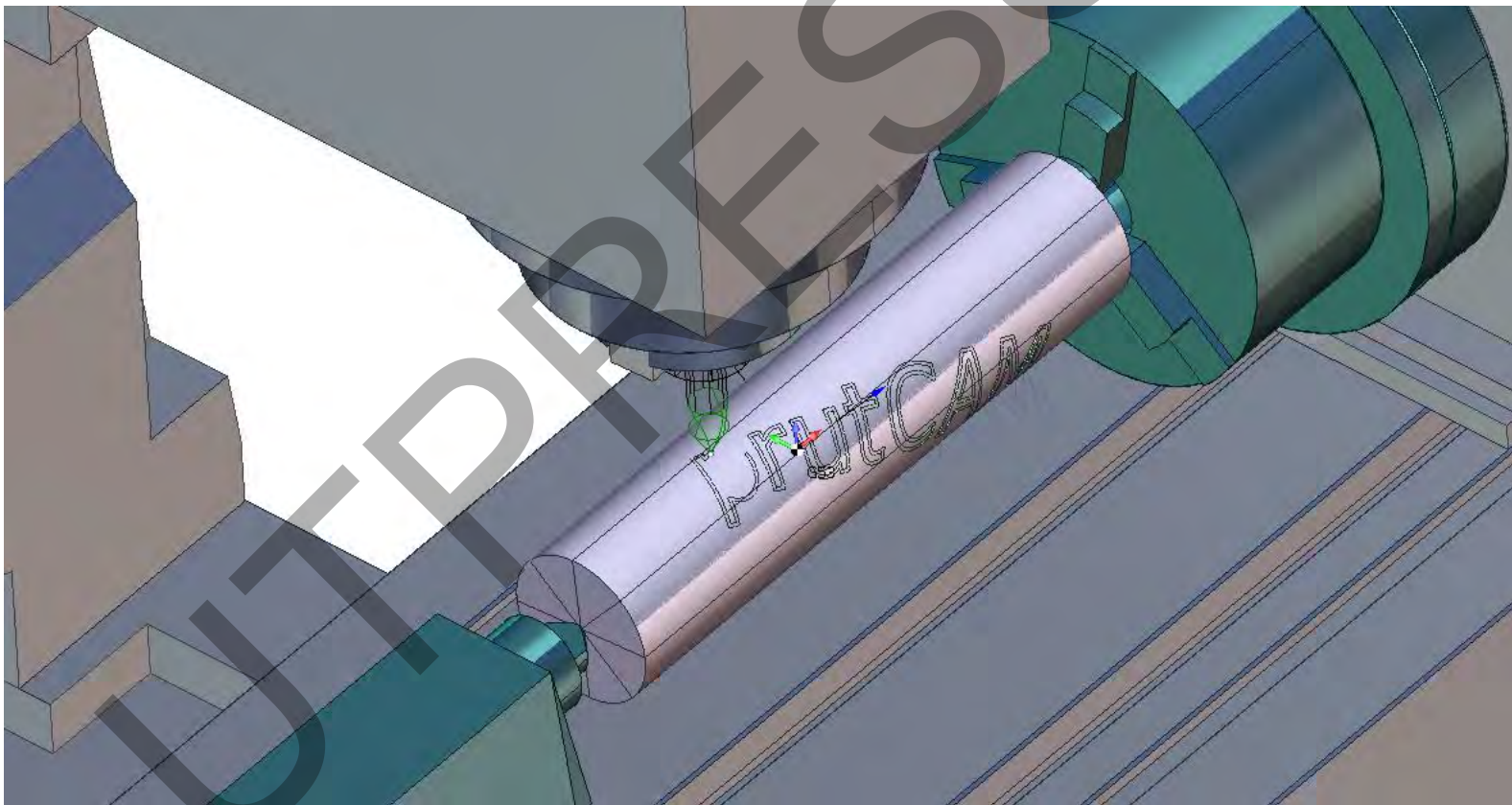
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Frezare continuă în 4 axe (X, Y, Z, A): simulare

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

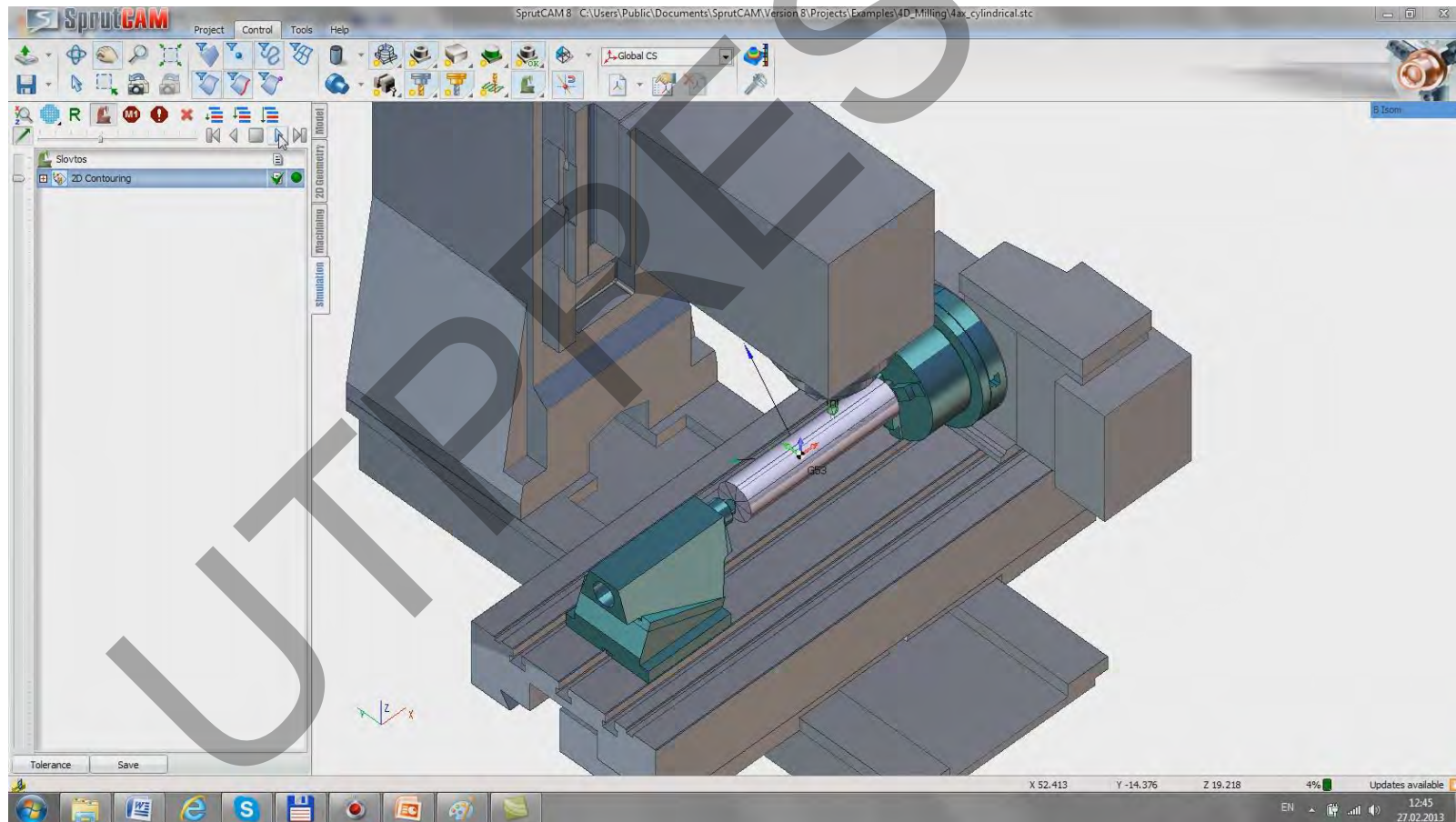
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

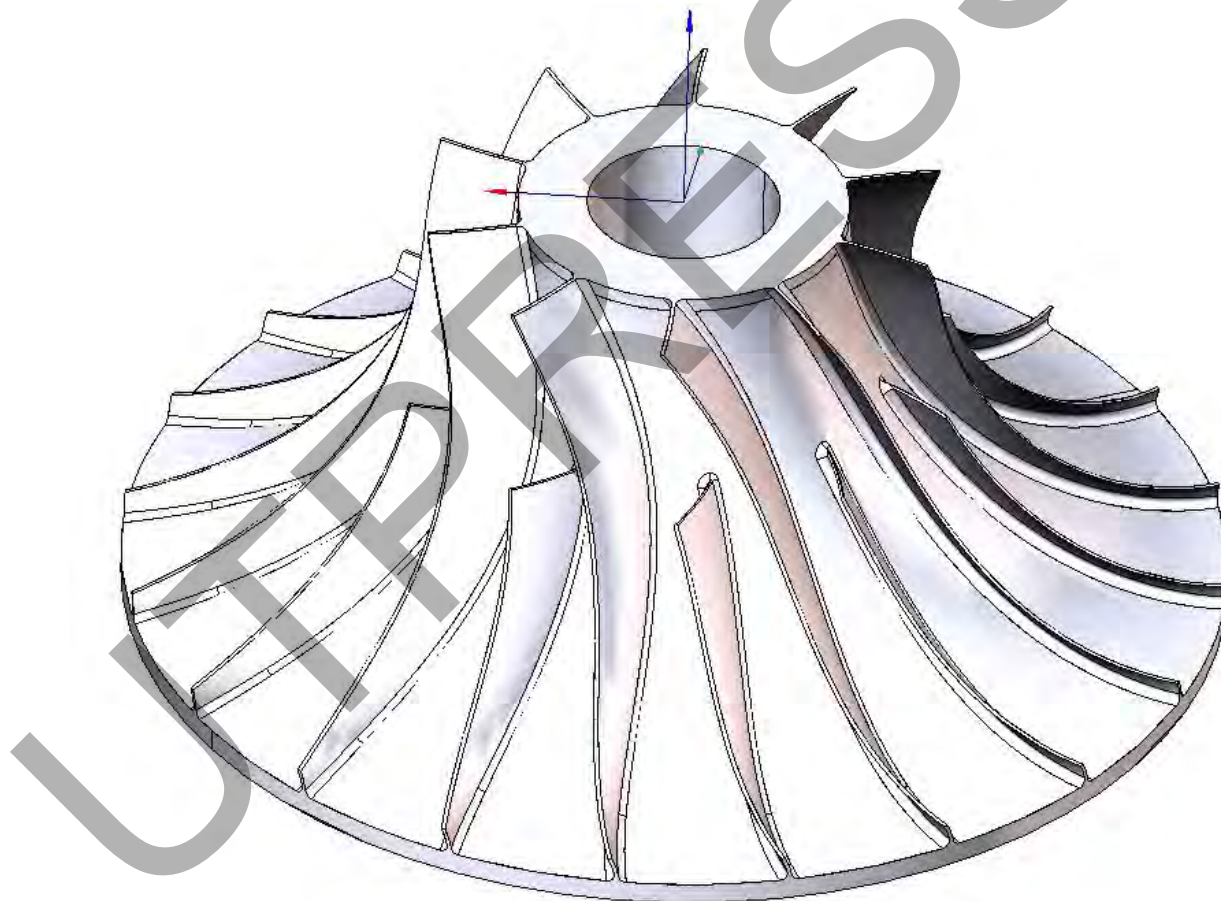
Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Frezare continuă în 5 axe (X, Y, Z, A, C) – modelul 3D al piesei



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

# Frezare continuă în 5 axe (X, Y, Z, A, C)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

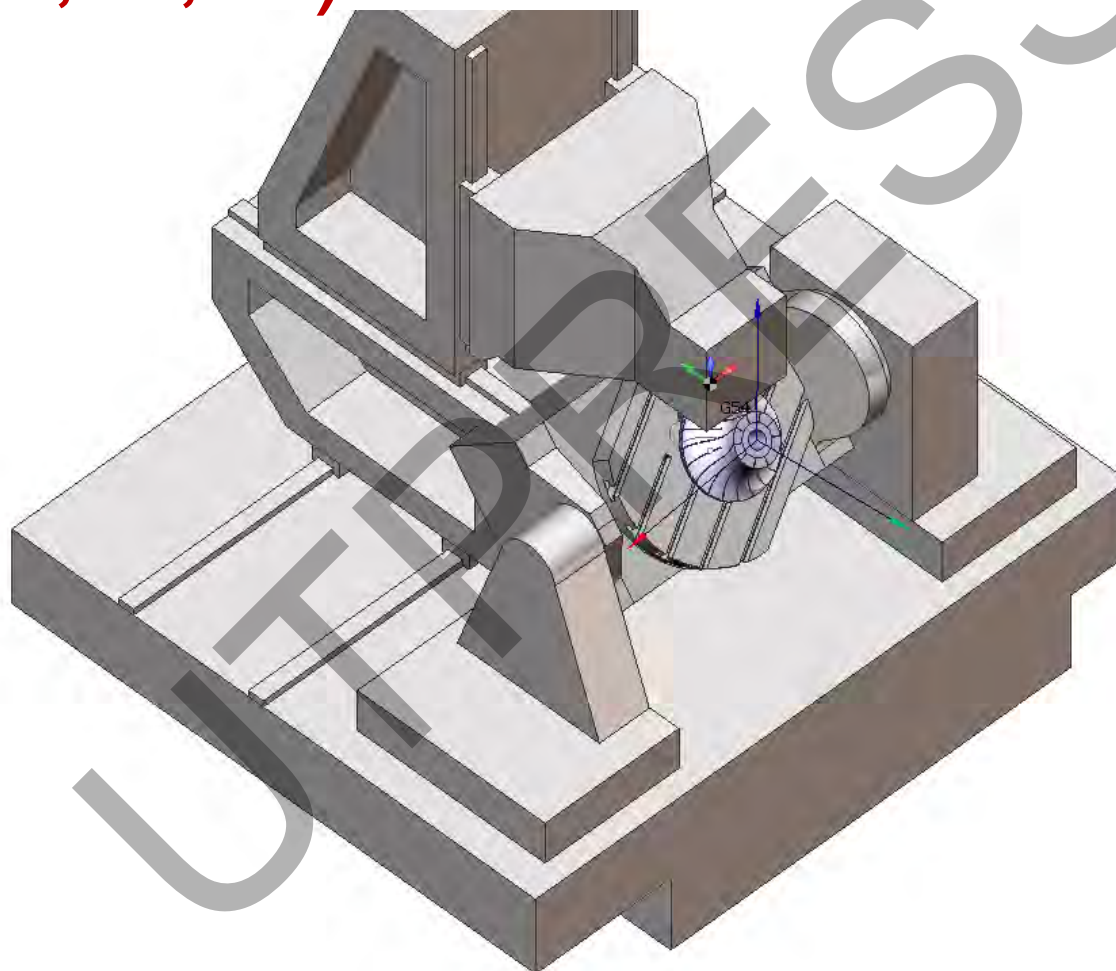
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



Se simulează  
și deplasările  
elementelor  
mobile ale  
mașinii

# Prelucrări combinate (frezare + strunjire)

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

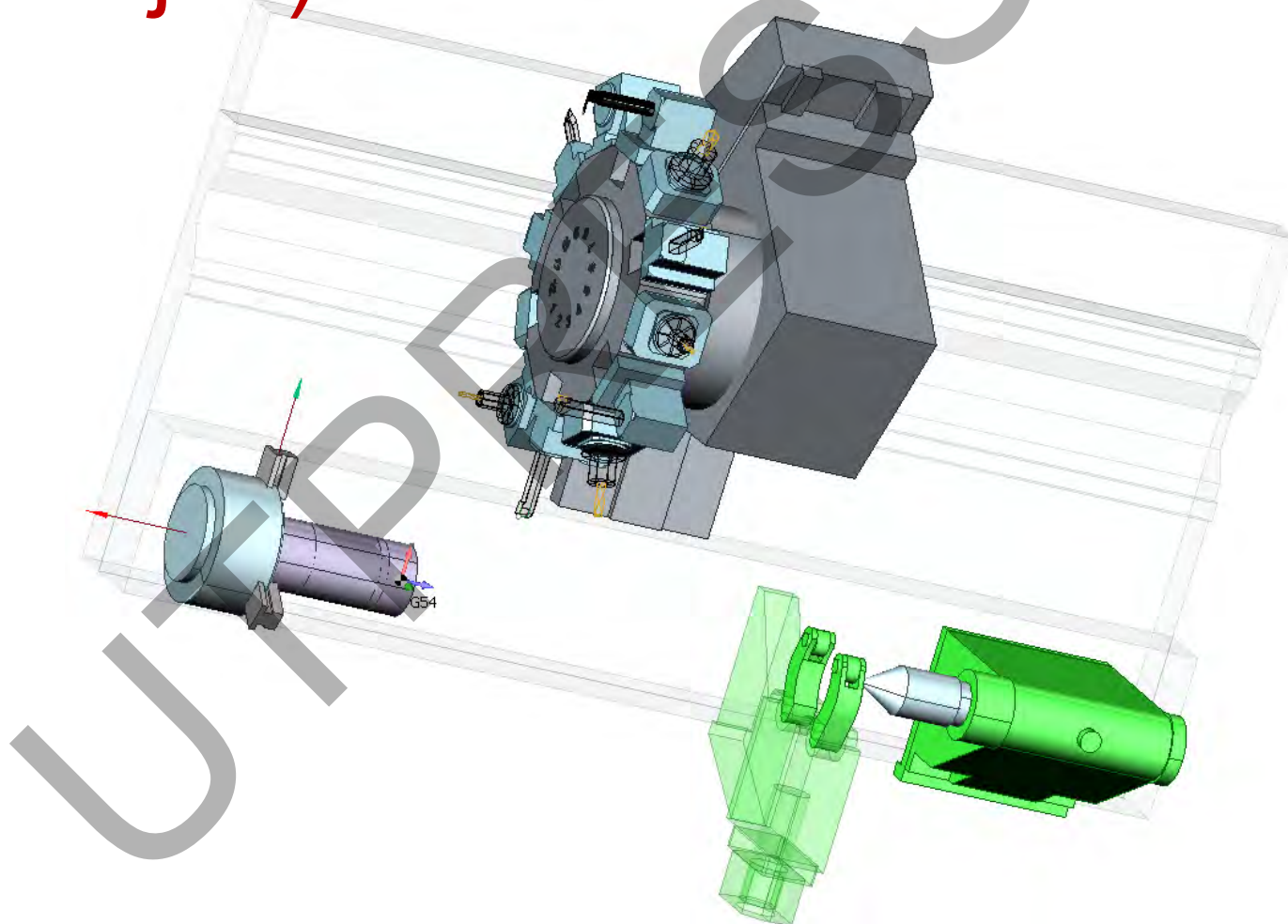
Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM



# Frezare cu roboți industriali seriali

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

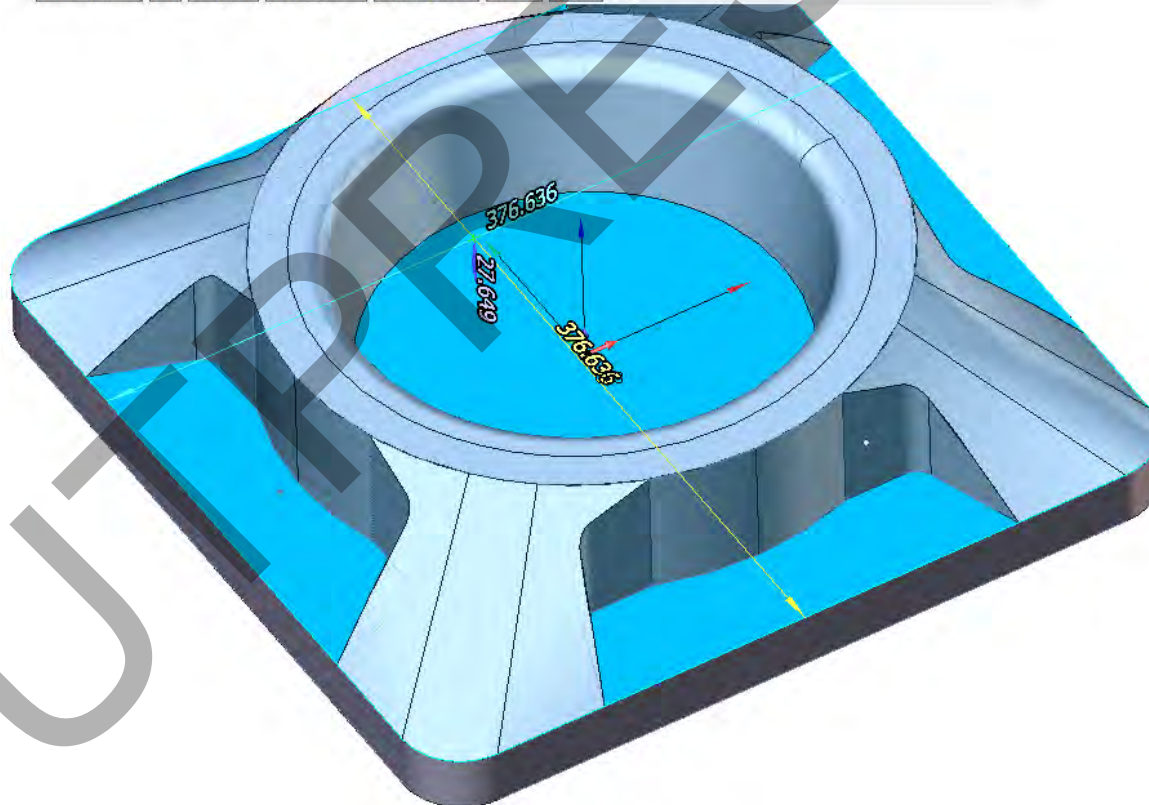
Cap.9. CAM



# Măsurarea dimensiunilor (pe modelul 3D al piesei)

Coordinates	Normal	Wall thickness	Gap amount	Length/Curvature
-3.384; 86.701; 0	0; 0; 1	27.649		Lu: 376.636; Lv: 376...
25.546; 99.948; 18....	-0.244; -0.955; 0.168			Ru: 103.161; Lv: 56....
93.528; 45.917; 24....	-0.885; -0.434; 0.168			Ru: 104.191; Lv: 56....

Geometry model / Part / Workpiece / Job assignment / Machining result / Fixtures / Options



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

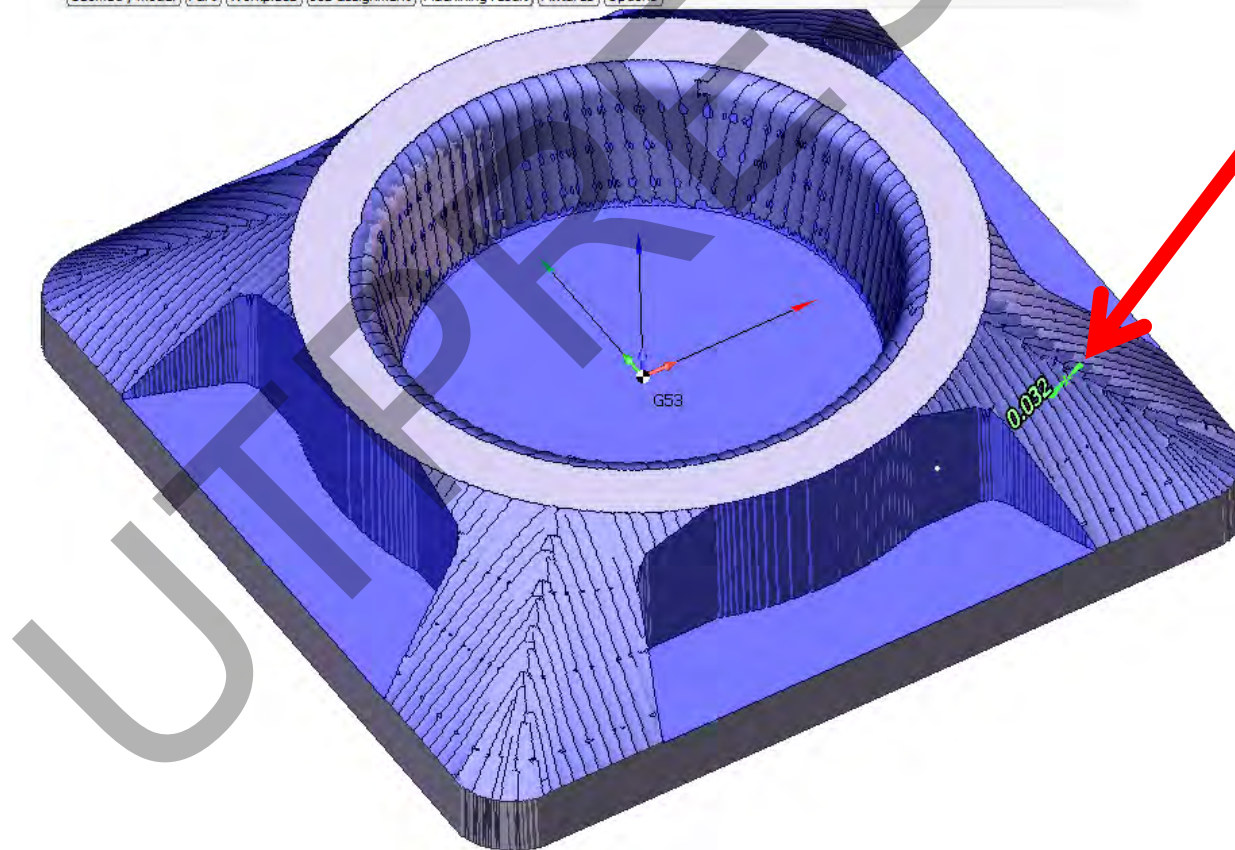
Cap.9. CAM



# Evaluarea dimensiunii adaosului pe piesa prelucrată

Pick	Stock	Coordinates	Normal	Wall thickness	Gap amount
Finishing drive(\$Body\$)String17(GOTO: X131.864, Y-144.9...)	0.032	124.237; -132.589; ...			
Finishing drive(\$Body\$)String255(GOTO: X142.177, Y-112....)	-0.003	129.471; -106.182; ...			
Finishing drive(\$Body\$)String17(GOTO: X131.864, Y-178.2...)	0	129.178; -163.69; 2...			

[\Geometry model](#) / [Part](#) / [Workpiece](#) / [Job assignment](#) / [Machining result](#) / [Fixtures](#) / [Options](#)



Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Cap.10. Încheiere

# Cap. 10. ÎNCHEIERE

Dacă după parcurgerea acestui suport de curs mai persistă îndoieli în întocmirea unui program CNC, sperăm ca paginile următoare să le elimine.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

Cap.10.Incheiere

## Câteva considerații generale:

Fiecare program întocmit trebuie să îndeplinească cel puțin două condiții:

- siguranța, are prioritate maximă;
- ușurința utilizării.

Obținerea unei siguranțe maxime în prelucrarea/realizarea preciziei impuse, evitarea rebuturilor, a coliziunilor, etc. poate fi obținută dacă programul în ansamblul său este gândit ca o succesiune de părți dedicate fiecărei scule în parte – tehnică similară cu utilizarea “blocurilor principale”.

Caracterul modal al unor informații, cum ar fi turația, avansul, etc, trebuie utilizat numai în cadrul blocurilor oferite prelucrării cu o anumită sculă. În acest fel, se evită apariția unor erori la reluarea programului după o oprire accidentală sau nu.

Studierea atentă a documentației și conlucrarea cu operatorul sunt alte premise ale întocmirii unui program corect și al materializării lui.

Programul propriu-zis are aceeași structură, chiar dacă ne raportăm la sisteme numerice de prelucrare diferite.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaaj ISO

Cap.4.  
Programarea  
deplasarilor

Cap.5.  
Informatii  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

Cap.10.Încheiere

În cazul centrelor de prelucrare prin frezare și strunjire, orice program este structurat pe patru secțiuni diferite:

1. Secțiunea de început program
  - prelucrare -
2. Secțiunea pentru dezactivarea sculei "i"
3. Secțiunea pentru activarea sculei "i+1"
  - prelucrare -
4. Secțiunea de sfârșit program

Ori de câte ori se întocmește un program nou, se va începe cu secțiunea de început program. Astfel, aproape se pot copia în întregime informațiile dintr-un program existent. Evident informații cum ar fi turația, avansul, scula, coordonatele vor fi altele, dar structura de bază rămâne aceeași ori de câte ori se începe un program .

După secțiunea de început a programului (prima sculă este activă) – urmează porțiunea de program destinată prelucrării.

După terminarea prelucrării cu prima sculă, în program urmează secțiunea a doua, destinată secvențelor de sfârșit în utilizarea primei scule. Imediat, după aceea urmează secțiunea de aducere în prelucrare a sculei următoare, urmată de secvențele de prelucrare cu această sculă nouă și structura acestor secțiuni rămâne aceeași, informațiile concrete specifice piesei, vor fi diferite.

Această basculare între secțiunile 2 și 3 se continuă până la activarea tuturor sculelor.

## Cap.1. Introducere

## Cap.2. Arhitectura unui program

## Cap.3. Limbaj ISO

## Cap.4. Programarea deplasărilor

## Cap.5. Informații tehnologice

## Cap.6. Cicluri fixe

## Cap.7. Subprograme

## Cap.8. Programarea strungurilor

## Cap.9. CAM

## Cap.10. Încheiere

După programarea informațiilor de prelucrare pentru ultima sculă, programul redactat se încheie cu secțiunea a patra.

Pentru exemplificare se va considera primul program prezentat în capitolul 3.

Analizând programul, prin prisma structurii prezentate anterior, se identifică:

- Secțiunea început program:

%

O 0001 ;

(număr program)

N0005 G91 G28 X0 Y0 Z0 ;

(deplasare în punctul de referință )

N0010 T01 M06 ;

(schimbare sculă, activare sculă T1)

N0015 G54 G90 S400 M3 ;

(setare zero piesă, programare în sistem absolut, pornire AP, cu 400 rot/min.)

N0020 G00 X-10.0 Y-15.0 ;

(deplasare în punctul de start)

N0025 G43 Z-5.0 D01 M8 ;

(poziționare pe Z, corecția de lungime, pornire, lichid de așchiere)

N0030 G01 Y65.578 F100 ;

(conturare, viteza de avans 100mm/min.)

Urmează prelucrarea conturului (informații specifice fiecărei prelucrări).

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8. Progra  
marea  
strungurilor

Cap.9. CAM

Cap.10. Incheiere

### -Secțiunea pentru dezactivarea sculei T1

N0085 M09 ; (oprire lichid așchiere)  
 N0090 G91 G28 G40 Z0 M19 ; (revenire în punctul de referință, anulare  
 corecție de lungime, oprire orientată AP)  
 N0095 M01 ; (stop opțional)  
 N0100 M05 ; (oprire AP)  
 N0105 T02 M06 ; (schimbare sculă, activare sculă T2)

### -Secțiunea pentru activarea sculei T2

N0110 G54 G90 S600 M3 ;  
 N0115 G00 X50.0 Y50.0 ;  
 N0120 G43 Z50.0 D02 M08 ;

Urmează prelucrarea cu scula T2.

Deoarece prelucrarea găurii presupune și utilizarea sculei T3 (burghiul) în program urmează în succesiune secțiunile doi și trei.

Cap.1.  
Introducere

Cap.2.  
Arhitectura  
unui program

Cap.3.  
Limbaj ISO

Cap.4.  
Programarea  
deplasărilor

Cap.5.  
Informații  
tehnologice

Cap.6. Cicluri  
fixe

Cap.7.  
Subprograme

Cap.8.Progra  
marea  
strungurilor

Cap.9.CAM

Cap.10.Încheiere

-Secțiunea de sfârșit program

```
N0150 M09 ;
N0155 G91 G28 G40 Z0 M19 ;
N0160 G28 X0 Y0 ;
N0165 M30;
%
```

Unele echipamente pot avea o structură a diferitelor secțiuni ușor modificată. Frecvent se recomandă ca în secțiunile de început program, activare sculă “i+1” să fie programate anumite condiții inițiale cum ar fi: anulare corecții, anulare cicluri, etc. Semnificația codurilor utilizate în program nu trebuie în acest moment să pună nici n fel de probleme pentru a le înțelege. Poate câteva precizări sunt bine venite în ceea ce privește G28. Comanda G28 este utilizată, atât la centrele de prelucrare prin frezare cât și prin strunjire, pentru a trimite mașina în punctul său de referință – un punct precis poziționat pe axele mașini, situat la extremitatea pozitivă a axelor. Este punctul care conferă siguranță în realizarea unor acțiuni cum ar fi schimbarea sculei. Pericolul unei eventuale coleziuni este complet exclus.

La centrele de prelucrare prin strunjire mai există un punct cu aproximativ același rol numit punctul de referință al sculei.

Modul de programare indicat ( utilizarea codului G91- programare în sistem incremental) este urmarea cerinței, prezentă la multe CNC-uri, de a trimite mașina via un punct intermediar – adică tocmai punctul în care se găsește. Utilizarea unui “pseudo - punct de referință”, chiar și în cazul echipamentelor care nu cer acest lucru, este de luat în seamă.