



## เอกสารอ้างอิง

### ภาษาไทย

1. ชวลิต รุ่งอิทธิวงศ์, "เทคนิคการควบคุมงานตามใบสั่งงาน," วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิศวกรรมอุตสาหกรรม วิศวกรรมศาสตรมหาวิทาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2529.
2. บุญชนะ อัดถากร, การบริหารกับคอมพิวเตอร์, สำนักพิมพ์โอเดียนสโตร์, 2515.
3. นิพนธ์ เล้าประจง, ระบบการควบคุมการผลิตเชิงวิศวกรรม, สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), พิมพ์ครั้งแรก, 2529.
4. ยืน ภู่วรรวน "A Case Study on N x M Job shop Scheduling Problem Using Heuristic Approach," วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิศวกรรมอุตสาหกรรม สถาบันเทคโนโลยีแห่งเอเชีย, 1980.
5. ริชาร์ด เจ. ซองเมอร์เกอร์, เทคนิคการผลิตแบบญี่ปุ่น, กรุงเทพฯ, ซีเอ็ดดูเคชั่น, บริษัท, 2529, หน้า 105 - 193.
6. สุรศักดิ์ นานานุกูล, การบริหารงานผลิต, สำนักพิมพ์ไทยวัฒนาพานิช จำกัด, กรุงเทพฯ, พิมพ์ครั้งที่ 2 (ปรับปรุงและแก้ไข), 2522.
7. อุดม ภูสวรรค์สิทธิ, "Optimal Production Scheduling for Iron Foundry Plant," วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิศวกรรมอุตสาหกรรม สถาบันเทคโนโลยีแห่งเอเชีย, 1984.

### ภาษาอังกฤษ

8. Blackburn, J.D. and Millen, R.A. "Improved Heuristics for Multistage-Requirements Planning System" Management Science, Vol 28, No.1, January 1972, pp 44-56.

9. Buffa, E.S., Modern Production / Operation Management, Seventh Edition, John Wiley & Sons; Inc., (1983)
10. Burbidge, John L., The Introduction of group Technology, Wiley, New York, 1975.
11. Cambell, H.G., Dudek, R.A., and Smith, M.L., "A Hearistic Algerithm for the n job m Machine Sequencing Problem," Management Science, Vol 16, No.10, June, 1970.
12. Clark, Frank J.et.al., Business System and Data Precessing Procedures, Prentice-Hall, Inc., Englewood Cliffs N.J. 1972.
13. Charles, Thomas Cullereth, Jr., "A Microcomputer Based Part Classification and Coding System to facilities Aided Manufacturing Applications in the Furniture Industrial," Ph.D. Department of Computer Engineering, University of North Carolina, 1986.
14. Chou Shun, See, "A Study of Line-Banlancing and Sequencing of a Multiple Product Assembly Line," M.Eng, AIT, 1987.
15. Dagramice, A., and Surkis, J., "Scheduling Independent Jobs on Parallel Indentical Processor," Management Science, Vol 25, No.12, December, 1979.
16. David D. Bedworth, James E. Bailey, "Integrated Production Control System, Management Analysis, Desing, John Wiley & Sons, 1982.
17. Echart, Reger. "A definition of Classification and Coding" Proceedings of CAM-I's Coding and Classification Workshop, Arlington, Texas, 1975.
18. Heany, D.F., (1968), "Development of Information System : What Management Needs to know, Ronald Press, New York

19. Houtzeel, Alexander. "An Introduction to the MICLASS System"  
Proceedings of CAM-I's coding and Classification Workshop,  
Arlington, Texas, 1975.
20. Jackson, J.R., "Scheduling a Production Line to Minimize Maximum  
Tardiness," research report No.43, Management Science  
Research Project, UCLA, January, 1955.
21. Johnson, S.M., "Optimal Two and Three-Stage Production Scheduling  
with Set up time Included," Navel Research Logistics  
Quarterly, Vol.1, No.1, March, 1954.
22. KHALID SIDDIQ, MUHAMMA "Multipul - project Multiple - Resource  
Constrained Scheduling," M.Eng, AIT, 1986.
23. Moore, J.M., "Sequencing n jobs on One Machine to Minimize the  
Number of Tary Jobs," Management Science, Vol.17, No.1,  
September, 1968.
24. Mudick, Robert G., and Joel F. Rase, (1971) Information System  
for Modern Management Prentice-Hall, Inc., Englewood  
Cliffs., N.J.
25. SHIGEO SHINGO ; "Study of "TOYOTA" Production System from  
Industrial Engineering Viewpoint," Japan Management  
Association, Tokyo Japan, 1981.
26. Sriskandarajah, Chelliah, "Production Model For a Sanitaryware  
Manufacturing Company," M.Eng, AIT, 1981.
27. Wilkerson, L.J., and Irwin, J.D., An Improved Method for  
Scheduling Independent Tacks, AIIE Tranactions, Vol.3,  
No.3, September, 1971.

ภาคผนวก

ภาคผนวก ก.

รายชื่อโรงงานเฟอร์นิเจอร์จากไม้ยางพาราในประเทศไทย

โรงงานทำเฟอร์นิเจอร์ไม้ยางพารา

ที่	รายชื่อบริษัท	ปริมาณใช้ไม้		ปริมาตรใช้ไม้/ปี 2530	จังหวัด
		ต่อเดือน	CU. FT.		
1.	บริษัทตอกเตอร์วู้ด จำกัด	30,000	360,000	10,194	อ.บางปะกง จ.ฉะเชิงเทรา
2.	บริษัทพนัสนิคมจำกัด	10,000	120,000	3,398	อ.พนัส ชลบุรี
3.	ทจก. ศรีราชาการช่าง	8,000	96,000	2,718	อ.ศรีราชา ชลบุรี
4.	ทจก. ตะวันออกกลาง	10,000	120,000	3,398	แกลง ระยอง
5.	บริษัทโตเกียวจำกัด	5,000	60,000	1,699	แกลง ระยอง
6.	บริษัทเอเชียจำกัด	50,000	600,000	16,990	แกลง ระยอง
7.	บริษัทนกแก้ววู้ดจำกัด	50,000	600,000	16,990	แกลง ระยอง
8.	บริษัทที่ พารา วู้ด จำกัด	30,000	360,000	10,000	แกลง ระยอง
9.	ร้านครุประสงค์	10,000	120,000	3,398	แกลง ระยอง
10.	บริษัทบี เอสพารา วู้ด จำกัด	40,000	480,000	13,592	ท่าใหม่ จันทบุรี
11.	บริษัทวู้ดแลนด์จำกัด	23,000	276,000	7,815	ปทุมธานี
12.	บริษัทแม่ใหญ่จำกัด	20,000	240,000	6,796	ปทุมธานี
13.	บริษัทไทยพงษ์จำกัด	15,000	180,000	5,097	อยุธยา
14.	บริษัทบางกอกเจริญมิตรจำกัด	15,000	180,000	5,097	กรุงเทพฯ
15.	บริษัทบางนาบ็อบบิลส์จำกัด	10,000	120,000	3,398	กรุงเทพฯ
16.	บริษัทแฟนซีวูอินตัสทรีจำกัด	20,000	240,000	6,796	กรุงเทพฯ
17.	บริษัทยมหอมจำกัด	15,000	180,000	5,097	กรุงเทพฯ
18.	บริษัทสยามธานีเฟอร์นิเจอร์	30,000	360,000	10,194	กรุงเทพฯ
19.	บริษัทสยามวู้ดจำกัด	25,000	300,000	8,495	กรุงเทพฯ

ที่	รายชื่อบริษัท	ปริมาณใช้ไม้ ต่อเดือน	ปริมาตรใช้ไม้/ปี CU. FT.	2530 CU. M.	จังหวัด
20.	บริษัทสินดีจำกัด	20,000	240,000	6,796	กรุงเทพฯ
21.	บริษัทอินเตอร์พาราว่า๊ดจำกัด	30,000	360,000	10,194	กรุงเทพฯ
22.	บริษัทวีแสง ไทยจำกัด	15,000	180,000	5,097	กรุงเทพฯ
23.	บริษัทเอส.พี. โฮม โปรดักส์	10,000	120,000	3,398	กรุงเทพฯ
24.	บริษัทยูนิบล็อกลินด์สทรีจำกัด	10,000	120,000	3,398	กรุงเทพฯ
25.	บริษัทสแกนไทยจำกัด	30,000	360,000	10,194	กรุงเทพฯ
26.	ทจก. เทพวิวัฒน์เฟอร์นิเจอร์	10,000	120,000	3,398	กรุงเทพฯ
27.	บริษัทสันไทยจำกัด	40,000	480,000	13,592	ยะลา
28.	บริษัทสยามว่า๊ดจำกัด	30,000	360,000	10,194	ตรัง
29.	บริษัทไทยพาราว่า๊ดจำกัด	40,000	480,000	13,592	ตรัง
30.	บริษัทสินดีจำกัด	20,000	240,000	6,796	ตรัง
31.	บริษัทวี. เควตจำกัด	20,000	240,000	6,796	(ใต้หวันเช่า)
32.	บริษัทไทยถาวรพาราว่า๊ดจำกัด	20,000	240,000	6,796	ยะลา
33.	บริษัทอุตสาหกรรมพาราว่า๊ด	15,000	180,000	5,097	
34.	บริษัทการุณีจำกัด	20,000	240,000	6,796	
35.	บริษัทนครศรีธรรมราชจำกัด	15,000	180,000	5,097	
36.	บริษัทพงษ์ไทยจำกัด	30,000	360,000	10,194	
37.	บริษัทหาดใหญ่พาราว่า๊ดจำกัด	20,000	240,000	6,796	
38.	บริษัทนัทลุงพาราว่า๊ดจำกัด	10,000	120,000	3,398	
	รวม	821,000	9,852,000	278,975	

ภาคผนวก ข.

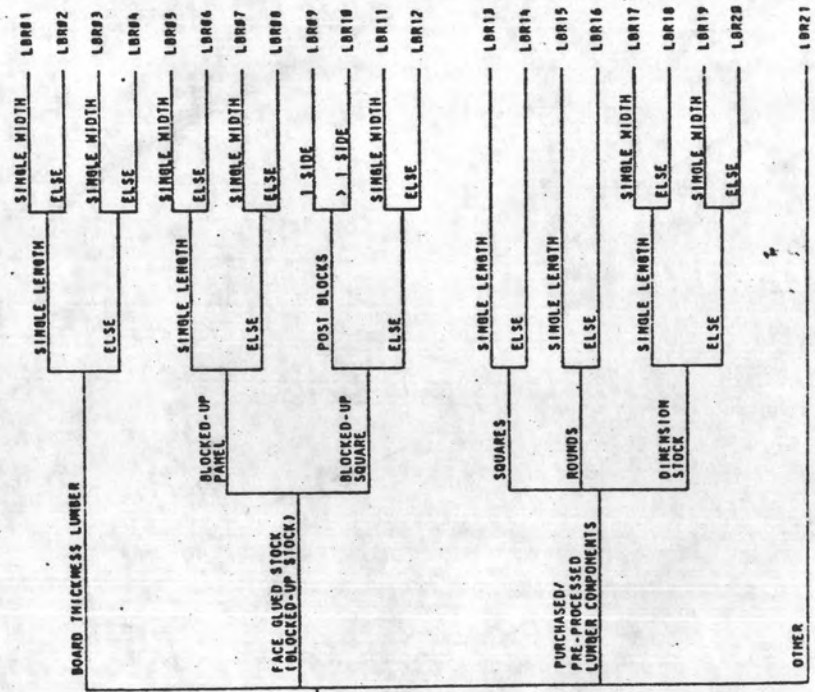
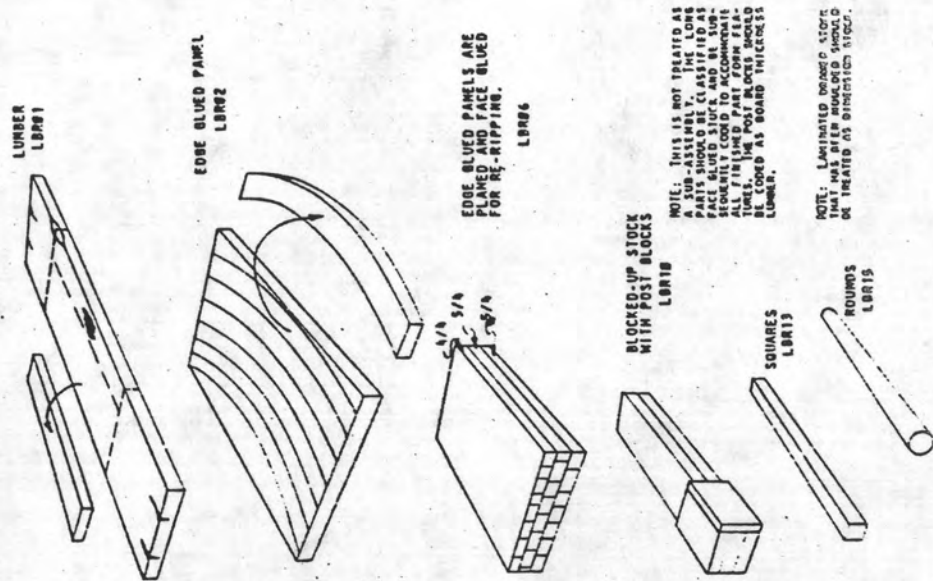
ระบบจำแนกและโคตชั้นส่วน

(Classification & Coding System)



MATERIALS TAXONOMIES

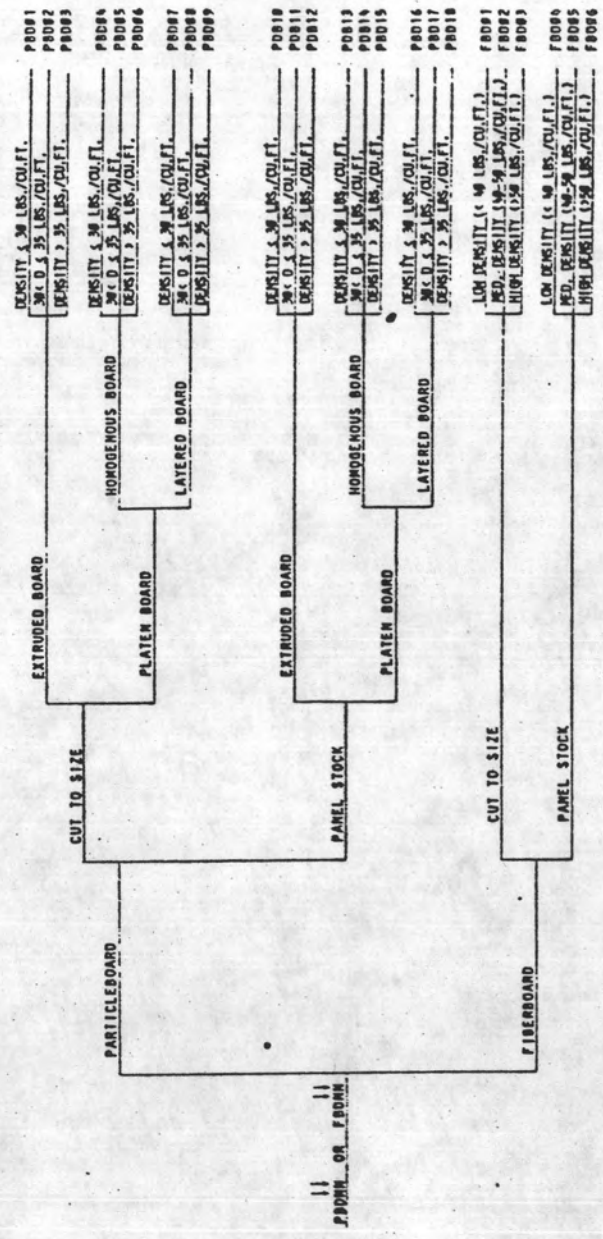
MATERIALS TYPHOGRAPH #1  
LUMBER AND WOOD PARTS



NOTE: THIS IS NOT TREATED AS A SUB-ASSEMBLY. THE LONG PARTS SHOULD BE CLASSIFIED AS SUCH. ALL FINISHED PARTS FOR FEASIBILITY OF THE POST BLOCKS SHOULD BE TREATED AS BOARD THICKNESS LUMBER.

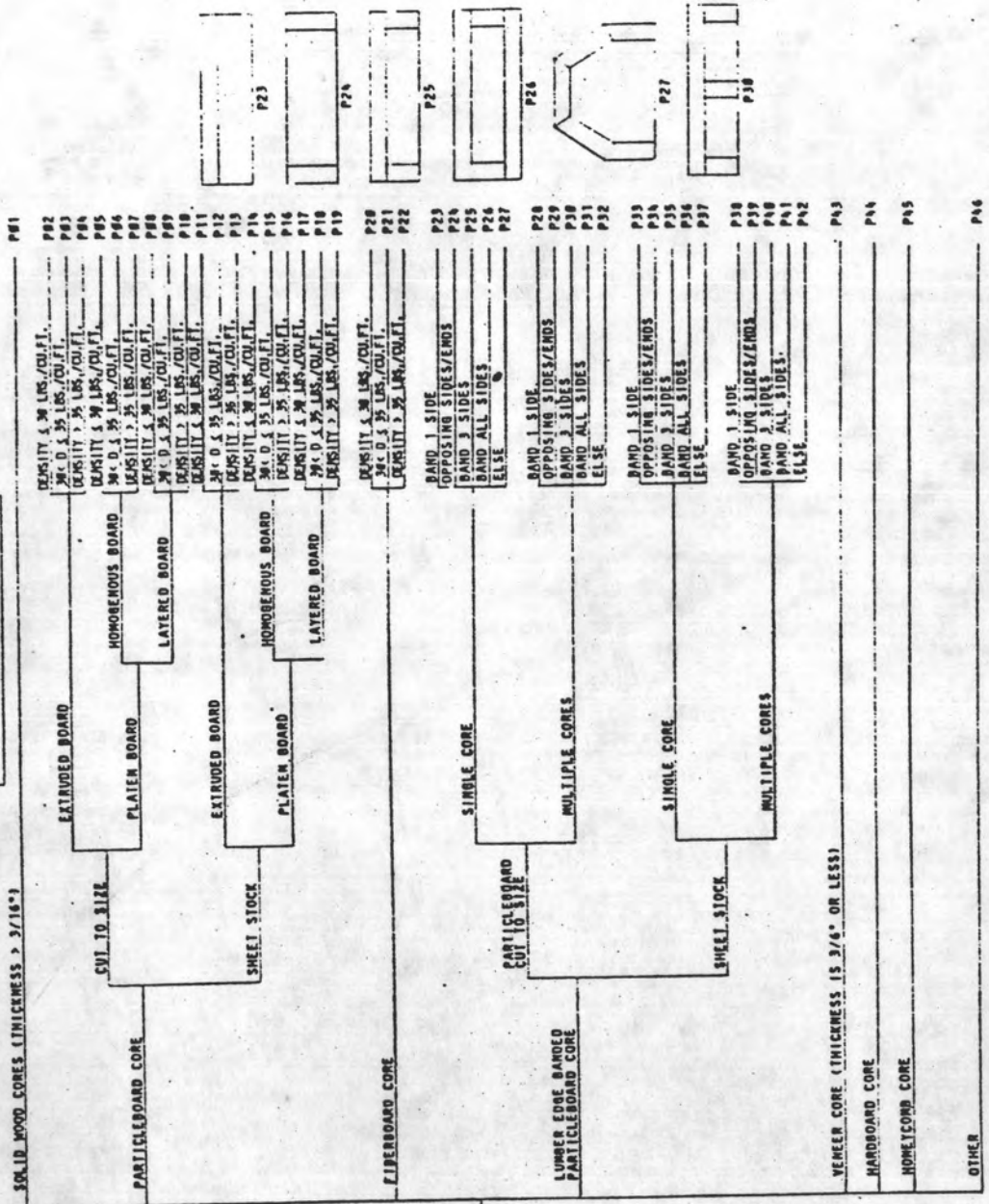
NOTE: LAMINATED BORED STOCK THAT HAS BEEN BORED SHOULD BE TREATED AS DIMENSION STOCK.

MATERIALS TAXONOMY #2  
PARTICLEBOARD & FIBERBOARD



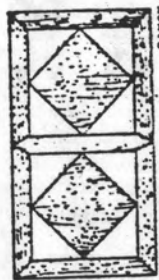
||  
||  
DOWN OR FROM

MATERIALS TAXONOMY #3-A  
FLAT PLYWOOD - SUBSTRATES



MATERIALS TAXONOMY #3-B  
 FLAT PLYWOOD - SURFACE MATERIALS

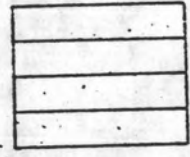
WOOD VENEERS	ASSEMBLED FACES (FANCY FACE PATTERN PRESENT)	3-PLY CONSTRUCTION	01
		3-PLY CONSTRUCTION	02
		7-PLY CONSTRUCTION	03
		OTHER	04
	ELSE	3-PLY CONSTRUCTION	05
		3-PLY CONSTRUCTION	06
		7-PLY CONSTRUCTION	07
		OTHER	08
		3-PLY CONSTRUCTION	09
		3-PLY CONSTRUCTION	10
		7-PLY CONSTRUCTION	11
		OTHER	12
		3-PLY CONSTRUCTION	13
		3-PLY CONSTRUCTION	14
		7-PLY CONSTRUCTION	15
		OTHER	16
	FACE AND BACK		17
HIGH PRESSURE LAMINATES (FORMICA, MICARTA, ETC.)	ELSE	3-PLY CONSTRUCTION	18
		ELSE	19
LOW PRESSURE LAMINATES (MELANINE & OTHER PAPERS)	FACE AND BACK	PRE-PRINTED PATTERN	20
		NEUTRAL SURFACE	21
	ELSE	PRE-PRINTED PATTERN	22
		NEUTRAL SURFACE	23
VINYL (AND OTHER THERMOPLASTICS)	FACE AND BACK		24
	ELSE		25
SPECIAL WOOD LAMINATES (PARQUET, ETC.)			26
OTHER			27



FANCY FACE  
 5-PLY  
 LUMBER CORE  
 P2102



FANCY FACE  
 3-PLY  
 FIBERBOARD CORE  
 P2101

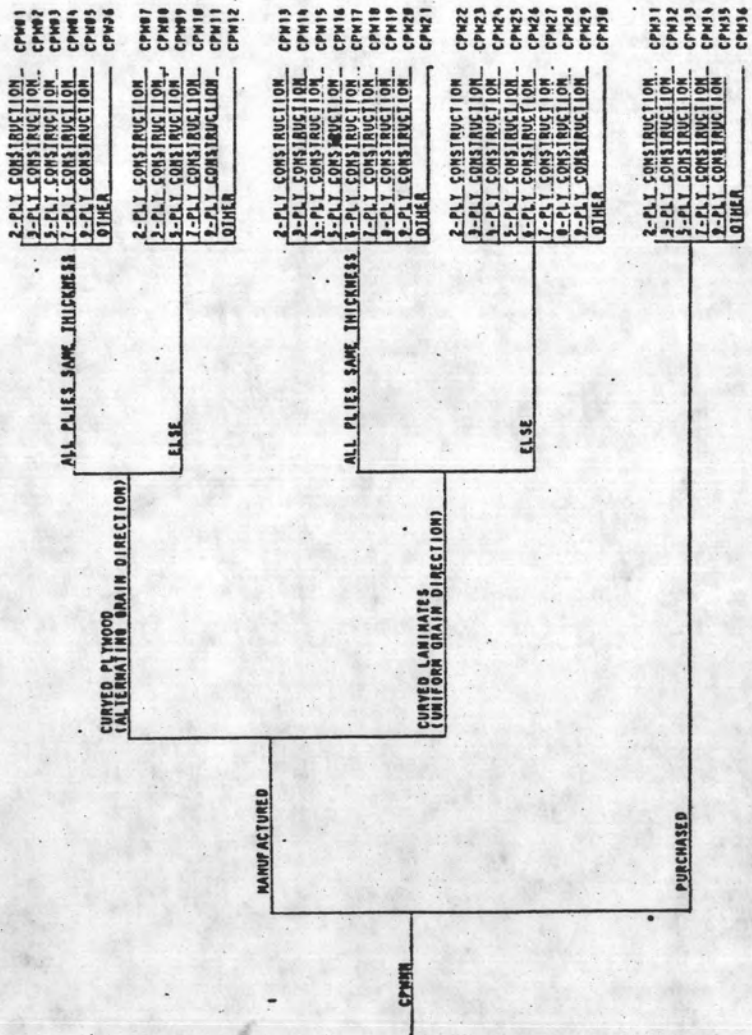


PLAIN FACE  
 3-PLY  
 PARTICLEBOARD CORE  
 P2099

NOTE: A FANCY FACE IS VOLVED BOTH PARALLEL JOISTS, THEREFORE, THIS IS NOT A FANCY FACE.

||  
 P2101

MATERIALS TAXONOMY #  
CURVED PLYWOOD



MATERIALS TAXONOMY #5  
MISCELLANEOUS MATERIALS

COMMERCIAL (CONSTRUCTION) PLYWOOD	M0001
LUAUN (PANELS) SHEET STOCK	M0002
PLASTIC COMPONENTS	M0003
RIGID URETHANE MOLDING	M0004
OTHER	M0005

MNNNN

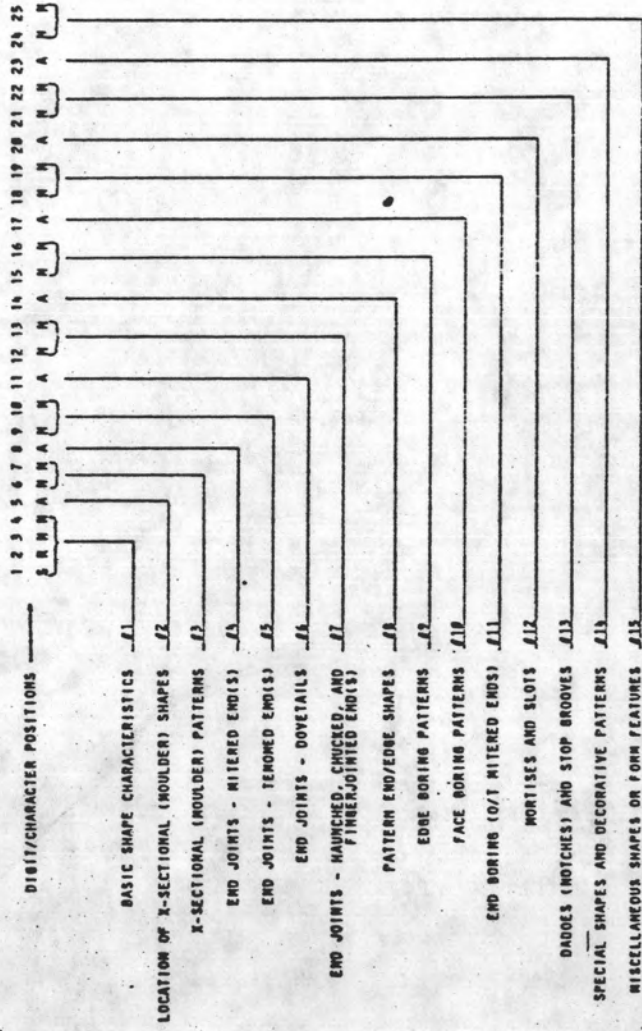
**PART GEOMETRY AND FORM FEATURES  
TAXONOMIES**

	NUMBER OF TAXONOMIES	NUMBER OF DIGIT/CHARACTER POSITIONS
1. STRAIGHT RAILS A. WIDTH $\leq 8"$ B. STRAIGHT AXIS EXISTS THRU PART LENGTH	15	25
2. FLAT PANELS A. WIDTH $> 8"$ B. NO CURVED PLANE EXISTS	12	20
3. TURNINGS A. PARTS INVOLVING ROTATIONAL MANUFACTURING PROCESSES (INCLUDES 3-D CARVING)	7	13
4. CURVED RAILS A. WIDTH $\leq 8"$ B. NO STRAIGHT AXIS EXISTS THRU PART LENGTH	12	20
5. CURVED PANELS A. WIDTH $> 8"$ B. AT LEAST ONE CURVED PLANE EXISTS	7	13
6. IRREGULAR PARTS A. ALL OTHER PART CONFIGUR- ATIONS	1	4
7. SUB-ASSEMBLIES A. AN ASSEMBLED CONFIGURATION OF COMPONENT PARTS THAT REQUIRE SUBSEQUENT MACHIN- ING OPERATIONS	11	19



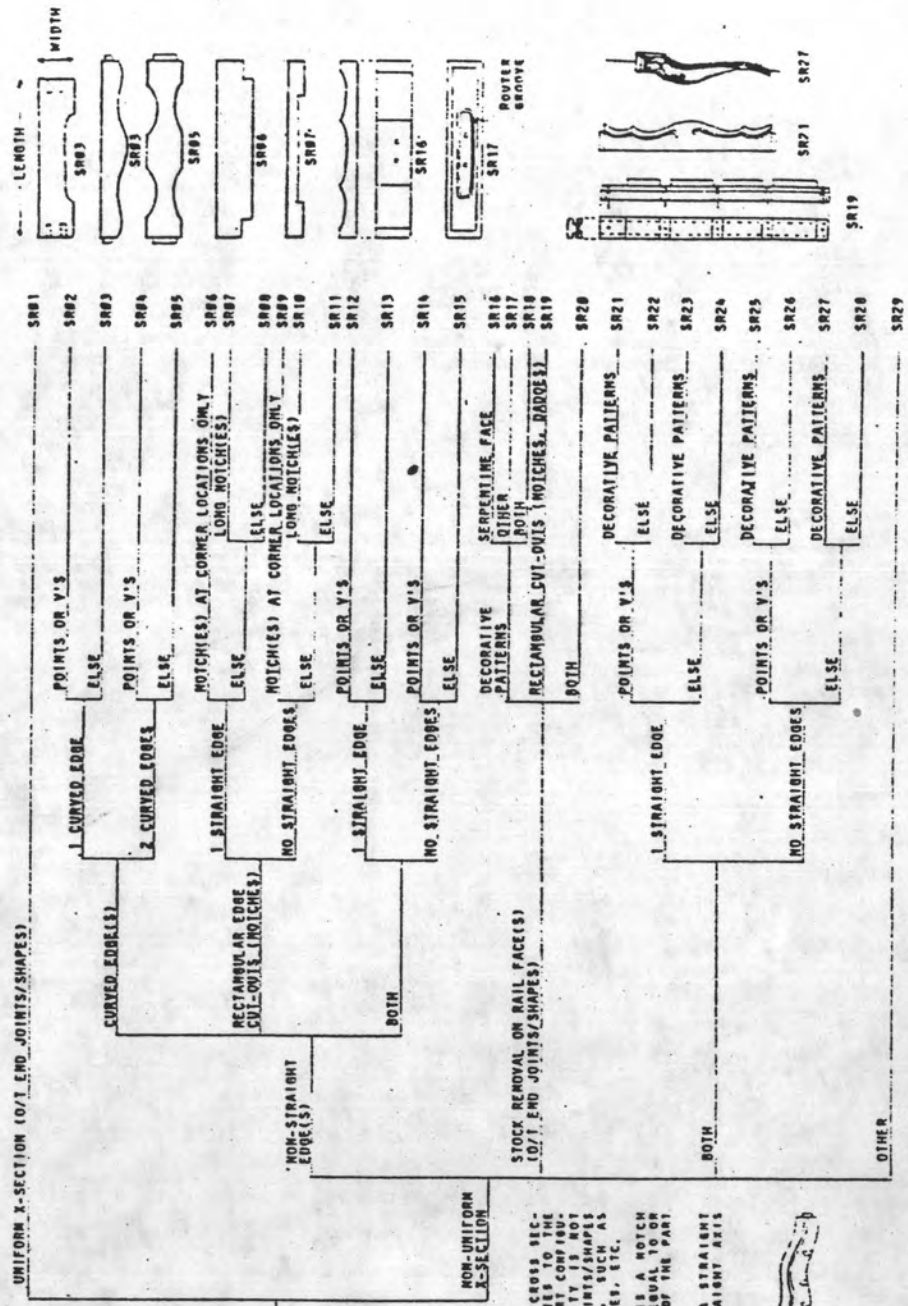
STRAIGHT RAIL TAXONOMIES

A = ALPHABETIC  
N = NUMERIC



**STRAIGHT RAIL TAXONOMY #1  
BASIC SHAPE CHARACTERISTICS**

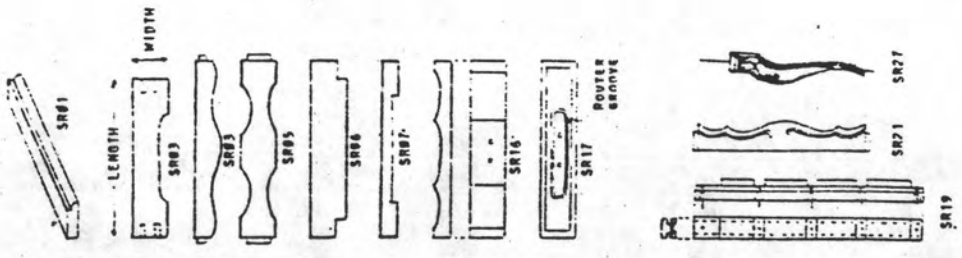
**NOTE:**  
A STRAIGHT RAIL IS A PART SUCH THAT:  
1. THE WIDTH IS 50° (FINISHED DIM.)  
2. A STRAIGHT EDGE EXISTS THROUGHOUT THE PART LENGTH (FINISHED DIM.)



**NOTE:** UNIFORMITY OF CROSS SECTIONAL SHAPE APPLIES TO THE FINAL (FINISHED) PART CONFIGURATION. UNIFORMITY IS NOT DISTURBED BY END JOINTS/SHAPES OR FORM FEATURES, SUCH AS BORED HOLES, NOTCHES, ETC.

**NOTE:** A LONG NOTCH IS A NOTCH THAT HAS A LENGTH EQUAL TO OR EXCEEDING ONE HALF OF THE PART LENGTH.

**NOTE:** THIS IS NOT A STRAIGHT RAIL SINCE NO STRAIGHT AXIS EXISTS.



STRAIGHT RAIL TAXONOMY #2  
LOCATION OF X-SECTIONAL  
(MOULDER) SHAPES

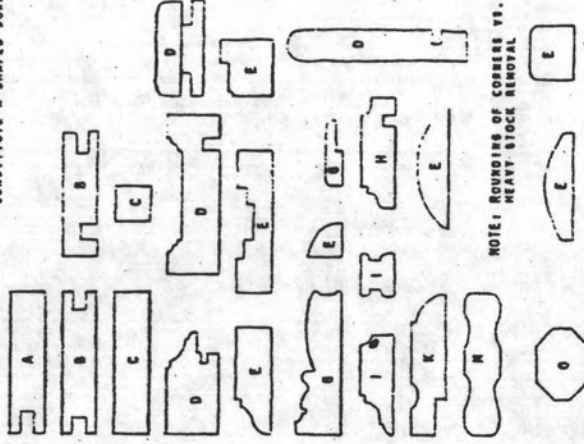
NOTE: THIS TAXONOMY DESIGNED TO REFLECT THOSE CROSS-SECTIONAL SHAPES THAT ARE PRODUCED ON A MOULDER OF SIMILAR MACHINE. THOSE SHAPES WHICH ARE PRODUCED BY OTHER MACHINES OR CURVED EDGES SHOULD NOT BE INCLUDED HERE.

SSS	A	SINGLE THRU BROOVE
	B	MULTIPLE THRU BROOVES
	C	ELSE
SHAPE 1 SURFACE	D	THRU BROOVING
	E	ELSE
SHAPE 2 ADJACENT SURFACES	F	THRU BROOVING
	G	ELSE
SHAPE 2 OPPOSING SURFACES	H	THRU BROOVING
	I	ELSE
SHAPE 3 SURFACES	J	THRU BROOVING
	K	ELSE
SHAPE 3 SURFACES	L	THRU BROOVING
	M	ELSE
SHAPE 3 SURFACES	N	THRU BROOVING
	O	ELSE

NOTE! THE FOLLOWING GUIDELINE SHOULD BE USED TO DETERMINE HOW MANY SURFACES OF A PART ARE SHAPED.  
CONSIDER THE FINISHED X-SECTIONAL SHAPE AND:  
1. DETERMINE THE NUMBER OF DISTINCT SURFACES REMAINING AFTER MOULDBING.  
2. COUNT ONLY THE REMAINING SURFACES WHICH HAVE SHAPES.

X-SECTIONAL SHAPES

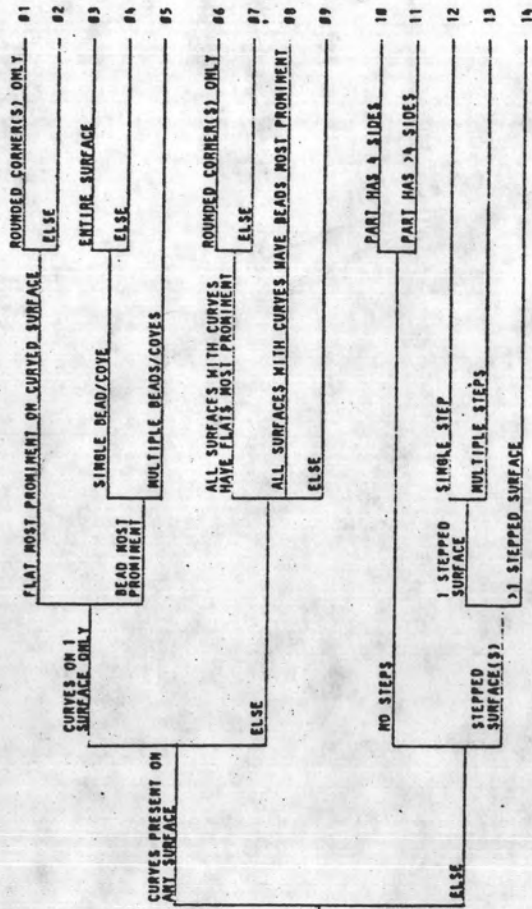
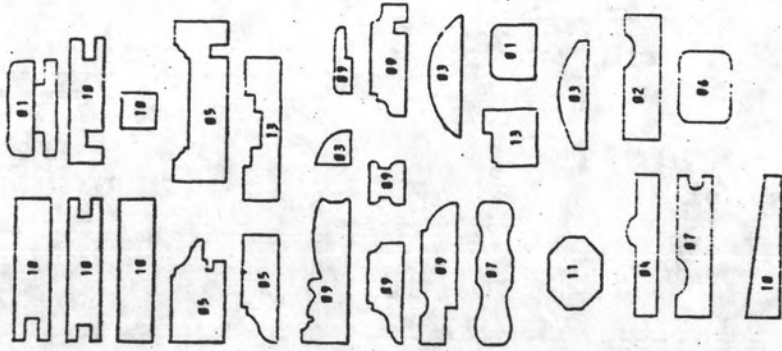
NOTE: BROOVING BY ITSELF DOES NOT CONSTITUTE A SHAPED SURFACE



NOTE: ROUNDING OF CORNERS VS. HEAVY STOCK REMOVAL



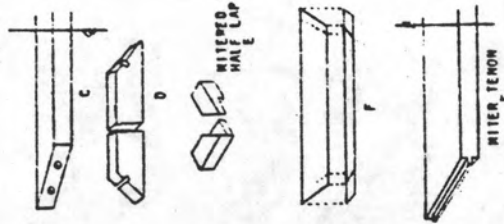
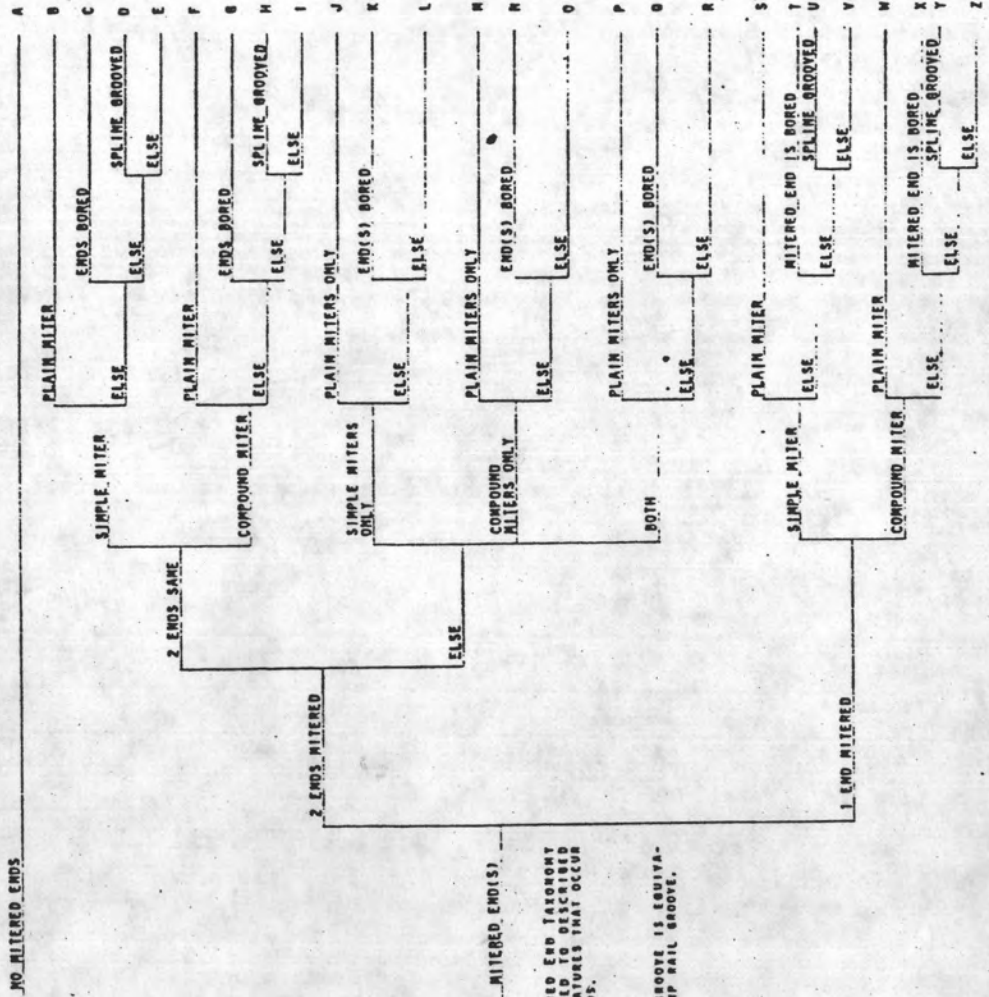
STRAIGHT RAIL TAXONOMY #3  
X-SECTIONAL (MOULDER) PATTERNS



NOTE: THE X-SECTIONAL (MOULDER) PATTERN TAXONOMY IS DESIGNED TO REFLECT THE DEGREE OF DIFFICULTY ENCOUNTERED IN THE SANDING OF STRAIGHT RAILS.

47  
BRUNN

STRAIGHT RAIL TAXONOMY (A)  
END JOINTS - MITERED ENDS

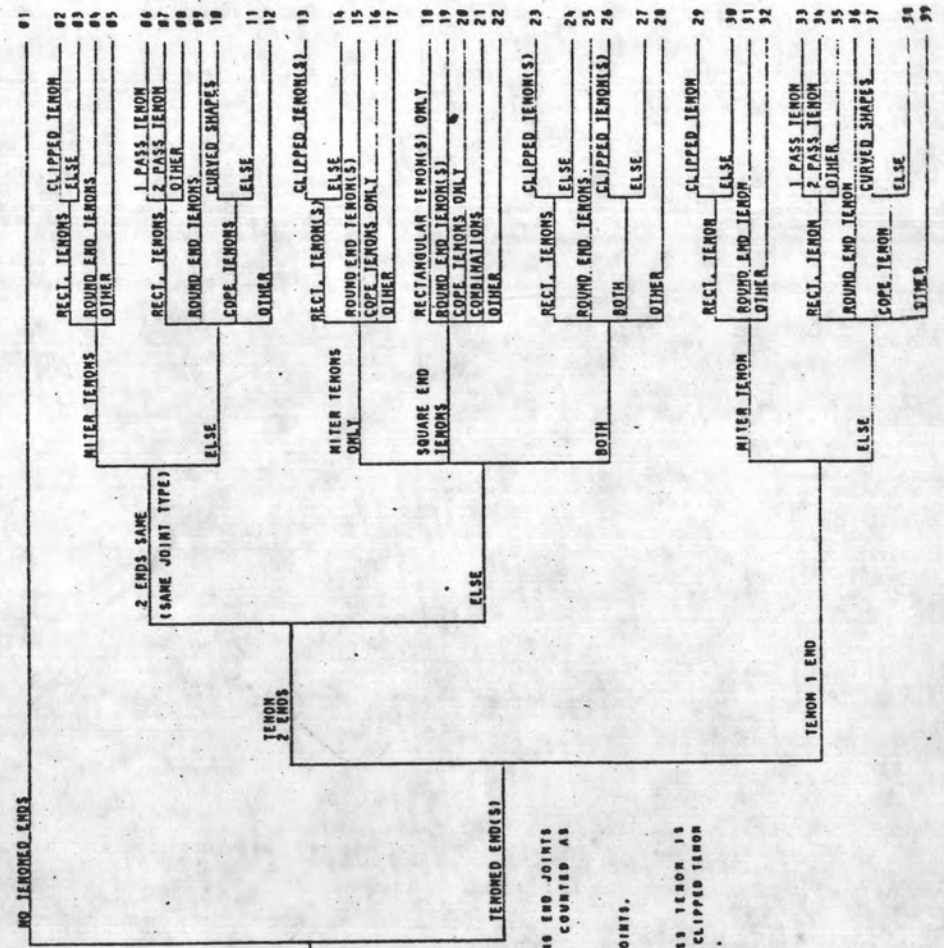
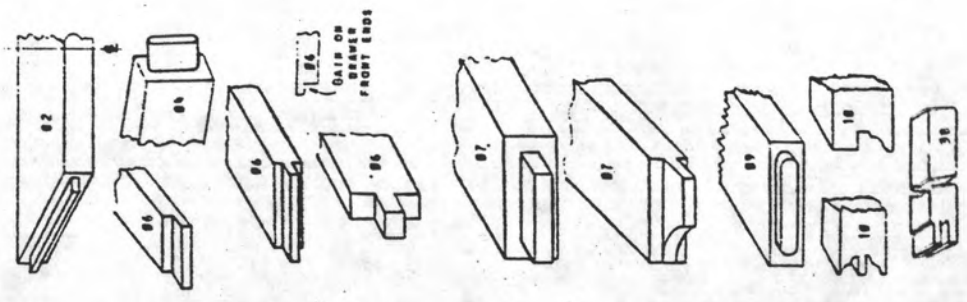


SRMANN

NOTE: THE MITERED END TAXONOMY SHOULD BE USED TO DESCRIBE ONLY THOSE FEATURES THAT OCCUR ON MITERED ENDS.

NOTE: A SPLINE GROOVE IS EQUIVALENT TO A CLAMP RAIL GROOVE.

STRAIGHT RAIL TAXONOMY #5  
END JOINTS - TENONED ENDS

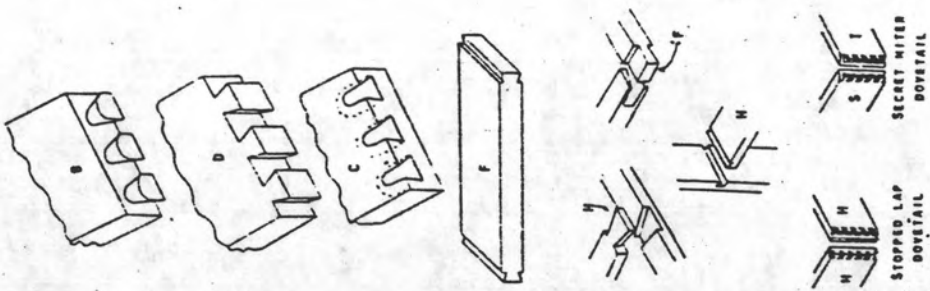
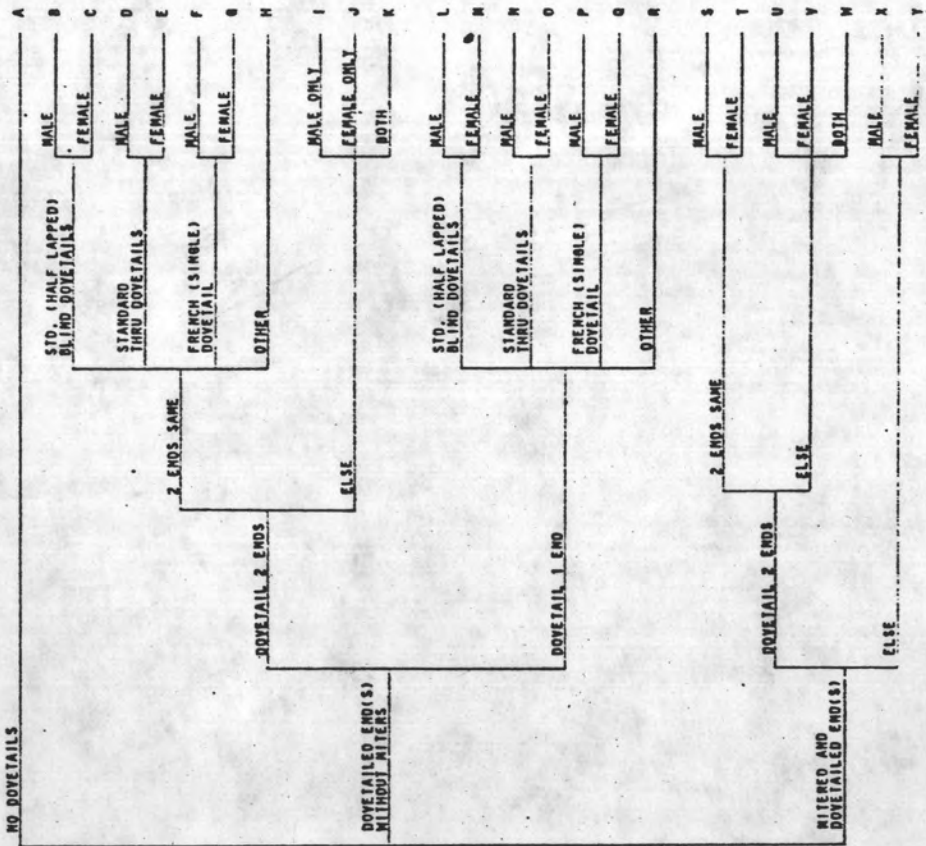


SPRINGER

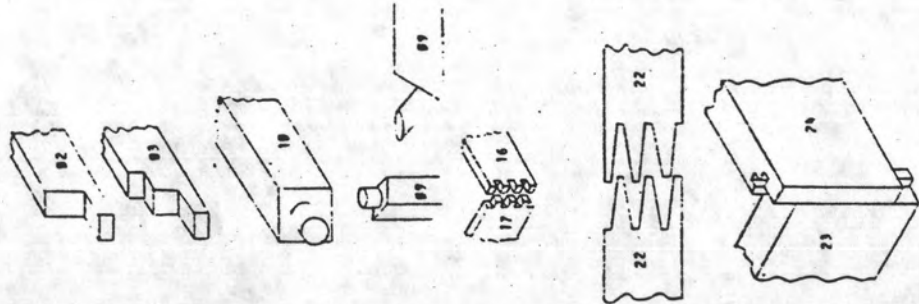
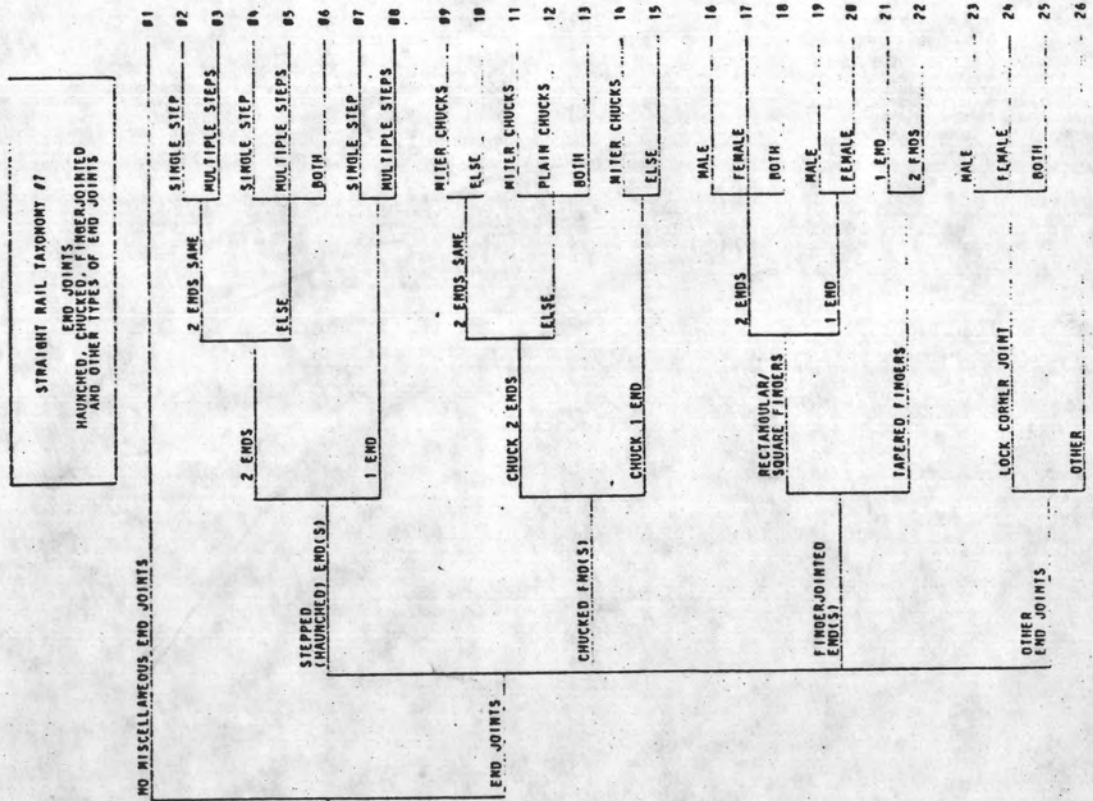
NOTE: THE FOLLOWING END JOINTS SHOULD NOT BE COUNTED AS TENONED JOINTS:  
1. FINGER JOINTS  
2. LOCK CORNER JOINTS.

NOTE: A TWO PASS TENON IS EQUIVALENT TO A CLIPPED TENON IN THIS TAXONOMY.

STRAIGHT RAIL TAXONOMY #6  
END JOINTS - DOVETAILS



11  
SAPPHIRADANA



12 13  
 SRINAWANN

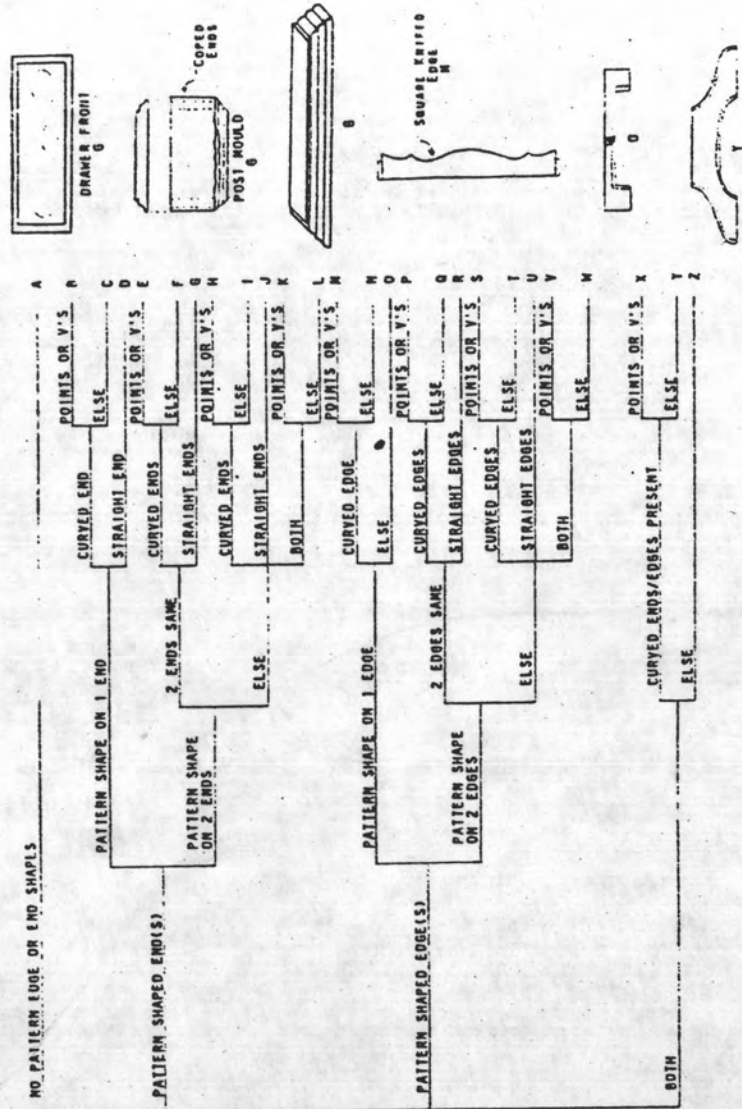


STRAIGHT RAIL TAXONOMY #8  
PATTERN EDGE/END SHAPES

NOTE: PATTERN EDGE/END SHAPES  
THE PATTERN SHAPES REFERRED TO  
IN THIS TAXONOMY OCCUR ON ENDS  
OR EDGES EXCLUSIVELY. ALSO,  
STRAIGHT EDGES THAT HAVE  
MOLDER SHAPES ARE NOT TO BE  
APPLIED TO THIS TAXONOMY.

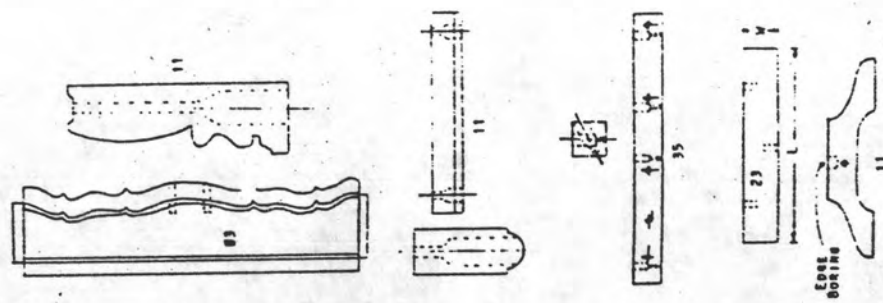
14  
SRINIVASAN

NOTE: PATTERN EDGE OR END SHAPES  
ARE PRIMARILY AESTHETIC. I.E.,  
NO EDGE OR END JOINTS ARE  
CREATED. THESE SHAPES ARE  
OBTAINED BY GRINDING THE  
PART. NORMALLY, A PATTERN  
EDGE SHAPE REQUIRES THE  
GRINDING OF A TOOL SPECIFIC TO  
THAT PATTERN. HOWEVER, SQUARE  
KNIFING OF RAIL ENDS OR EDGES  
IS CONSIDERED PATTERN SHAPING.



STRAIGHT RAIL TAPERMENT #9  
EDGE BORING PATTERNS

NO. EDGE BORING REQUIRED	81				
CURVED EDGE	ALL HOLES SAME DIAMETER	82	MIN. C-C SPACING .1"		
	ELSE	83	MIN. C-C SPACING .1"		
	ELSE	84	MIN. C-C SPACING .1"		
BORE_1_EDGE	85				
	86	MIN. C-C SPACING .1"			
	87	MIN. C-C SPACING .1"			
ELSE	88	MIN. C-C SPACING .1"			
	89	MIN. C-C SPACING .1"			
	90	MIN. C-C SPACING .1"			
CURVED EDGES ONLY	91	MIN. C-C SPACING .1"			
	92	MIN. C-C SPACING .1"			
	93	MIN. C-C SPACING .1"			
STRAIGHT EDGES ONLY	94	MIN. C-C SPACING .1"			
	95	MIN. C-C SPACING .1"			
	96	MIN. C-C SPACING .1"			
BOTH	97	MIN. C-C SPACING .1"			
	98	MIN. C-C SPACING .1"			
	99	MIN. C-C SPACING .1"			
OPPOSING_EDGES	100	MIN. C-C SPACING .1"			
	101	MIN. C-C SPACING .1"			
	102	MIN. C-C SPACING .1"			
ADJACENT_EDGES	103	MIN. C-C SPACING .1"			
	104	MIN. C-C SPACING .1"			
	105	MIN. C-C SPACING .1"			
BORE_2_EDGES	106	MIN. C-C SPACING .1"			
	107	MIN. C-C SPACING .1"			
	108	MIN. C-C SPACING .1"			
OTHER	109	MIN. C-C SPACING .1"			
	110	MIN. C-C SPACING .1"			
	111	MIN. C-C SPACING .1"			



15 16  
--SUNNANMANNAH--

NOTE: A RAIL NORMALLY HAS 2 EDGES, WITH THE THICKNESS DIMENSION CONSTITUTING AN EDGE. IF THIS IS NOT THE CASE FOR SQUARE RAILS (W/T = 1), 4 EDGES EXIST.



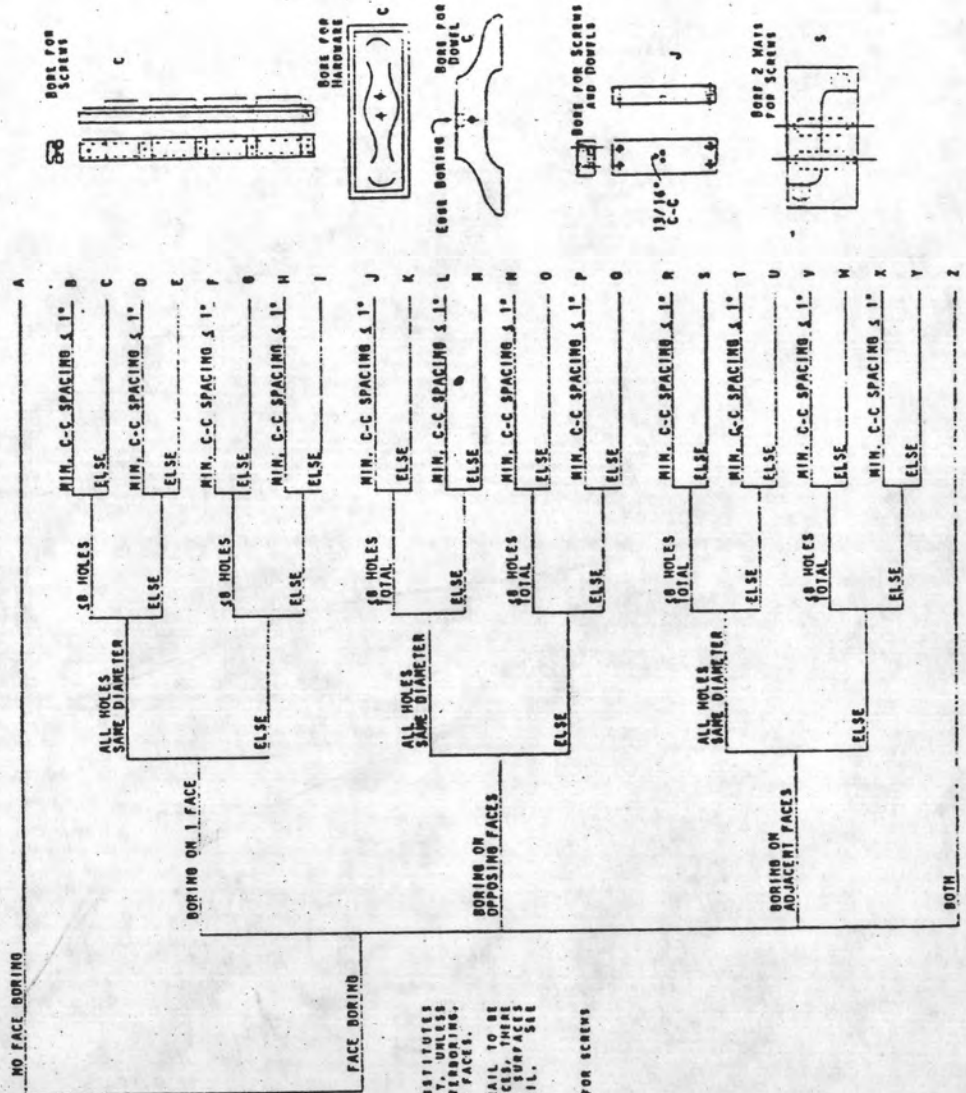
IF W/T = 1, 4 EDGES EXIST

HEXAGONAL RAILS HAVE 6 EDGES



ALSO, IF A RAIL HAS MORE THAN 4 SURFACES, RAILS, THEN IT HAS AS MANY EDGES AS IT HAS SURFACES.

STRAIGHT RAIL TAXONOMY #10  
FACE BORING PATTERNS



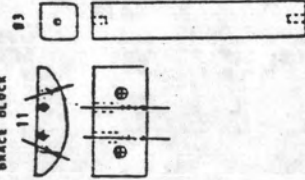
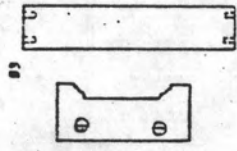
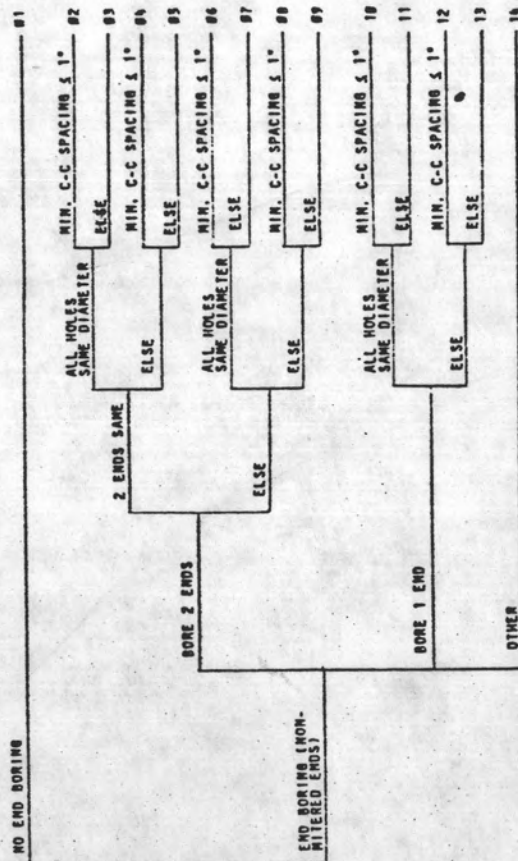
17  
SERVING AS A MARKER

NOTE: THRU BORING CONSTITUTES BORING ON 1 FACE ONLY, UNLESS COUNTERSINKING, COUNTERSINKING, ETC., ARE PRESENT ON 2 FACES.

NOTE: IN ORDER FOR A RAIL TO BE BORED ON 2 ADJACENT FACES, THERE MUST BE MORE THAN 4 SURFACES PRESENT ON THE RAIL. SEE EXAMPLE BELOW.

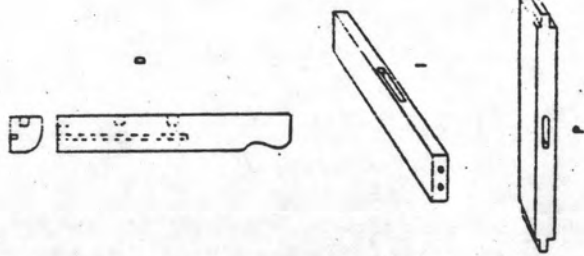
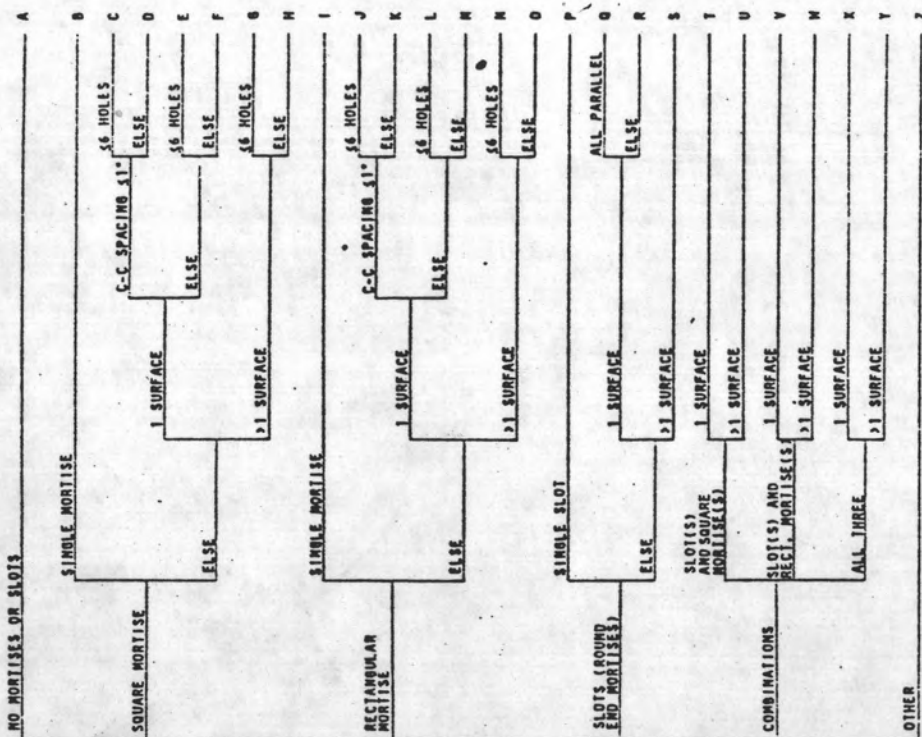


STRAIGHT RAIL TAXONOMY #11  
 END JOINTS - END BORING (O/T MITERED ENDS)



1819  
 V  
 SRRHHHHHHHHHHHH

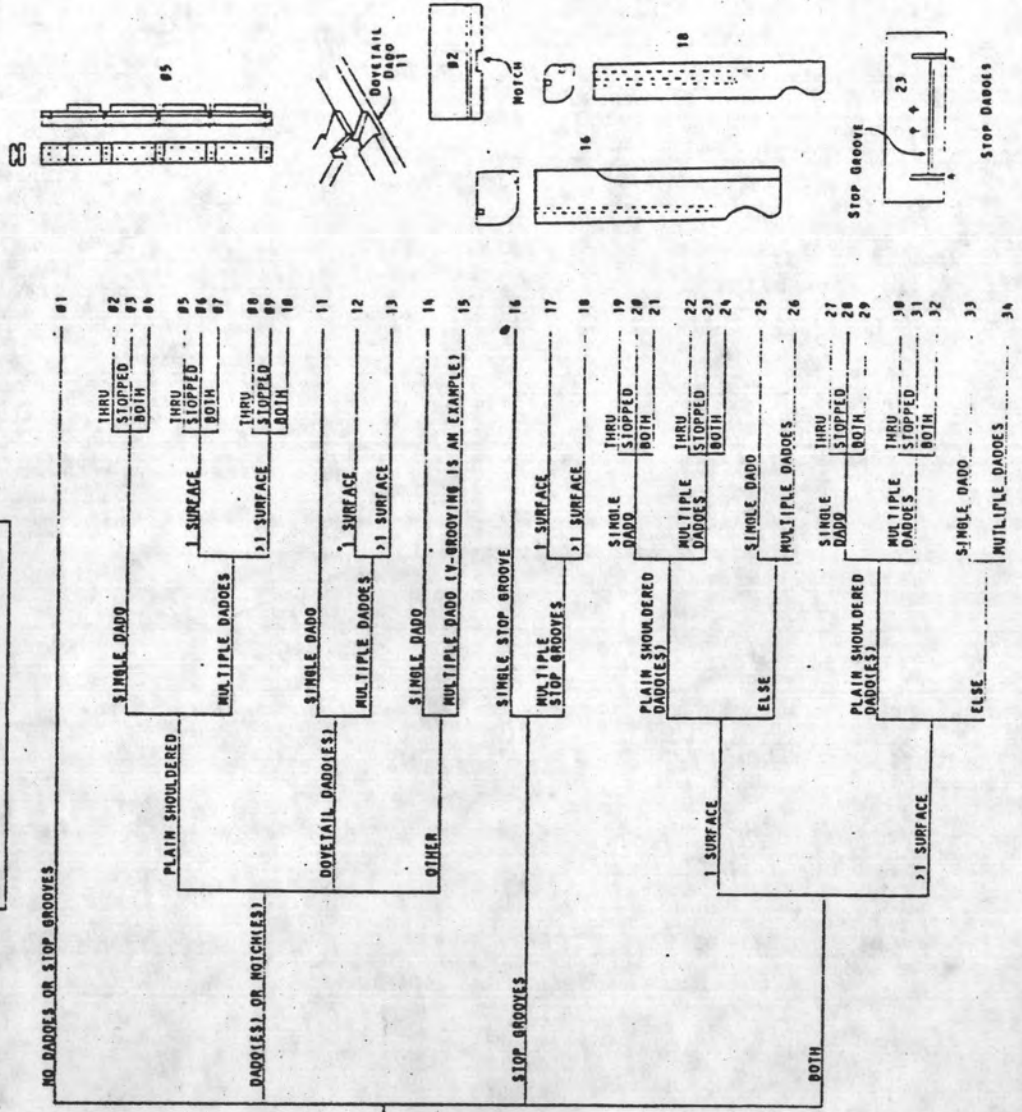
STRAIGHT RAIL TAIORORY #12  
FORM FEATURES - MORTISES AND SLOTS



20  
SRINIVASAN

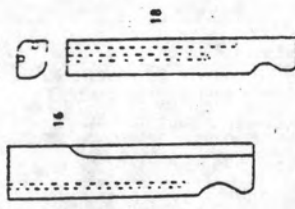
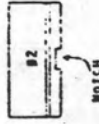
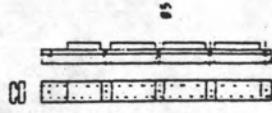
NOTE: A SLOT IS A MORTISE WITH ROUND ENDS. IT DIFFERS FROM MORTISES IN THAT A STOP GROOVE IS ALWAYS PULLED OUT TO AN EDGE. A SLOT IS CONFINED WITHIN THE SURFACE ON WHICH IT OCCURS AND IS EITHER PARALLEL OR PERPENDICULAR TO THE GRAIN.

STRAIGHT RAIL TATORYRY #13  
FORM FEATURES  
DADDOS (NOTCHES) AND STOP GROOVES



21 22  
SRHHNHNHNHNHNHNHNHNHN

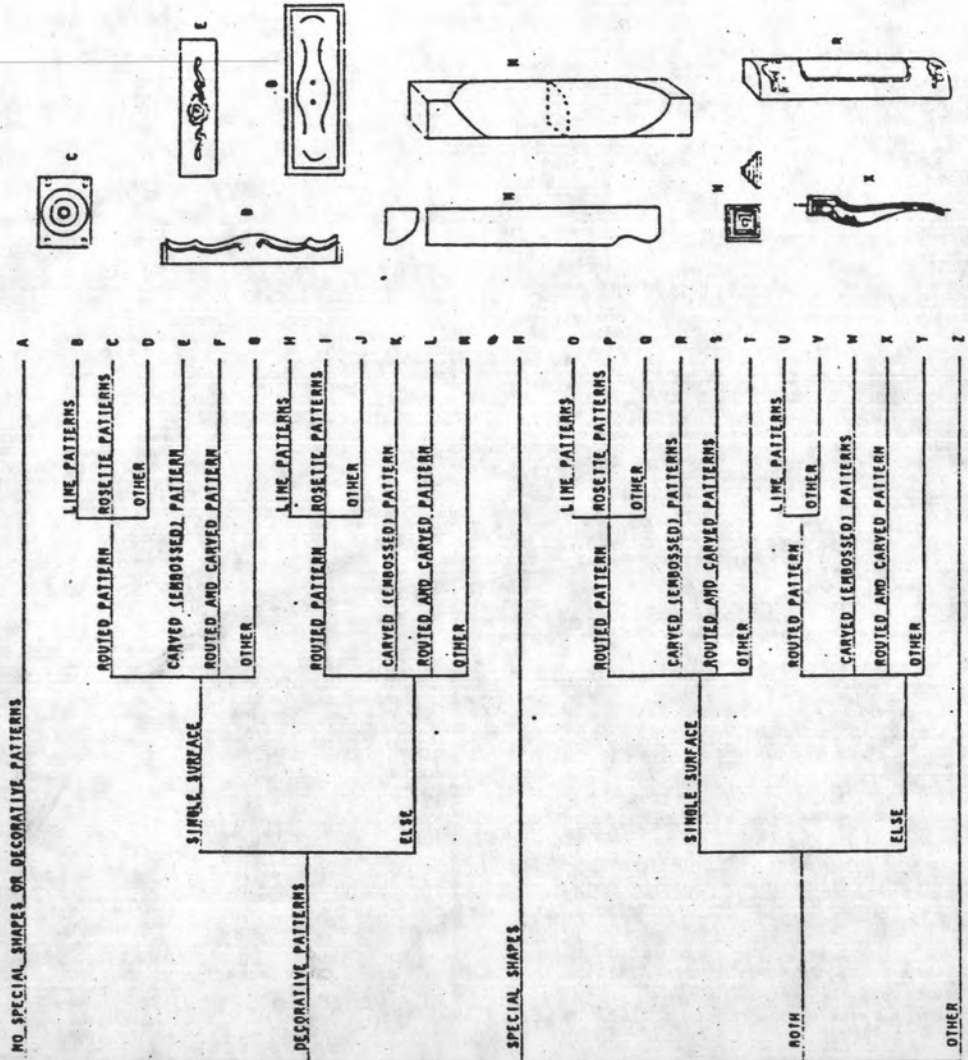
NOTE: A DADO MUST RUN IN A CROSS GRAIN DIRECTION.  
NOTE: A STOP GROOVE MUST BE PULLED OUT TO AN EDGE, OTHERWISE IT IS A SLOT.  
NOTE: A NOTCH IS EQUIVALENT TO A DADO. NORMALLY, A NOTCH OCCURS ON THE EDGE OF A RAIL, WHILE A DADO OCCURS ON THE FACE OF A RAIL.



STOP GROOVE

STOP DADDOS

STRAIGHT RAIL TAXONOMY FIN  
SPECIAL SHAPES AND DECORATIVE PATTERNS



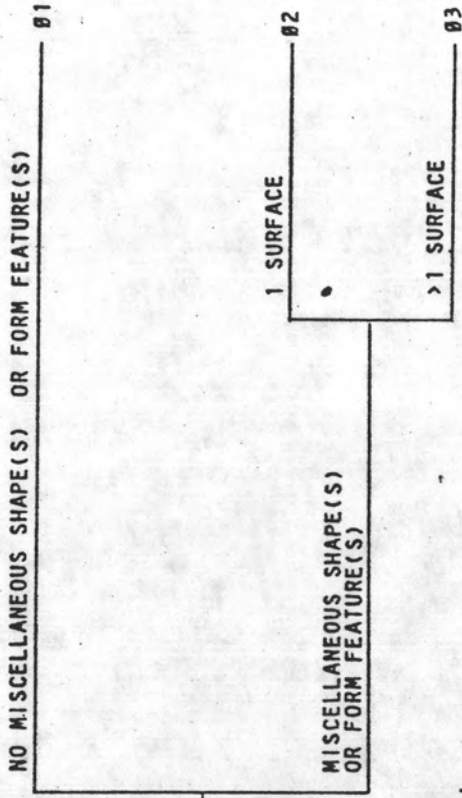
NOTE: DECORATIVE PATTERNS ARE ORNAMENTAL FEATURES THAT DO NOT CHANGE THE OVERALL DIMENSIONS. HOWEVER, SOME CARVING AND/OR EMBOSSED PATTERNS MAY RESULT IN DIMENSIONAL CHANGES.

23  
SRHANNHANNHANNHANNH

NOTE: A SPECIAL SHAPE IS ONE THAT IS NOT STRICTLY CONSIDERED TO BE A SPECIAL END AS PER TAXONOMY 08.

NOTE: IF ALL SPECIAL DECORATIVE FEATURES AND SHAPES ARE TO BE CARVED ONTO A RECTANGULAR BLANK, THEN IT SHOULD BE CLASSIFIED AS EITHER R OR X.

STRAIGHT RAIL TAXONOMY #15  
MISCELLANEOUS SHAPES OR FORM FEATURES



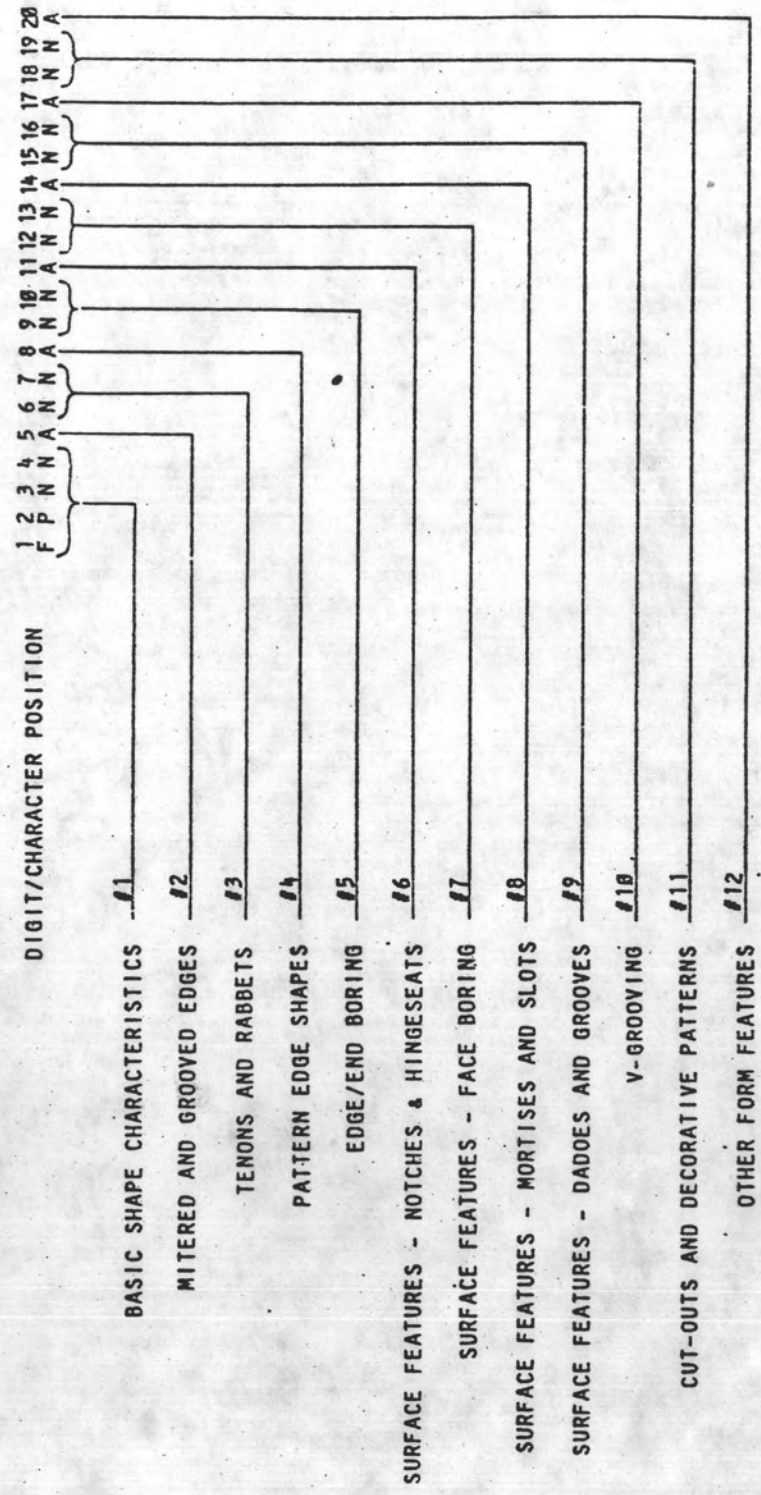
24 25

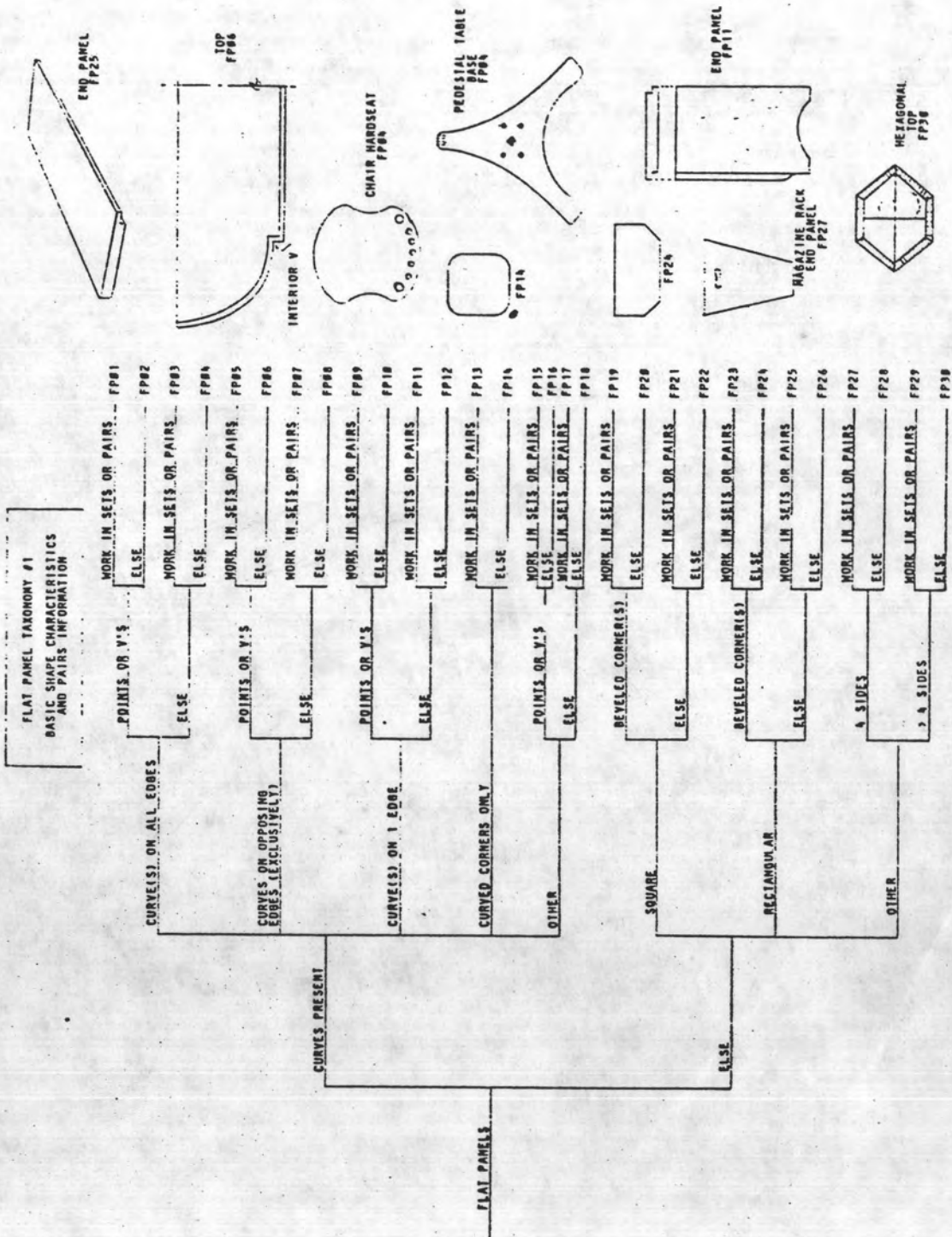
SRNNANNANNANNANNANNANN



FLAT PANEL TAXONOMIES

NOTE: A = ALPHABETIC  
N = NUMERIC





NOTE: ALL GROOVES ARE PLAIN  
RECTANGULAR CUTS.



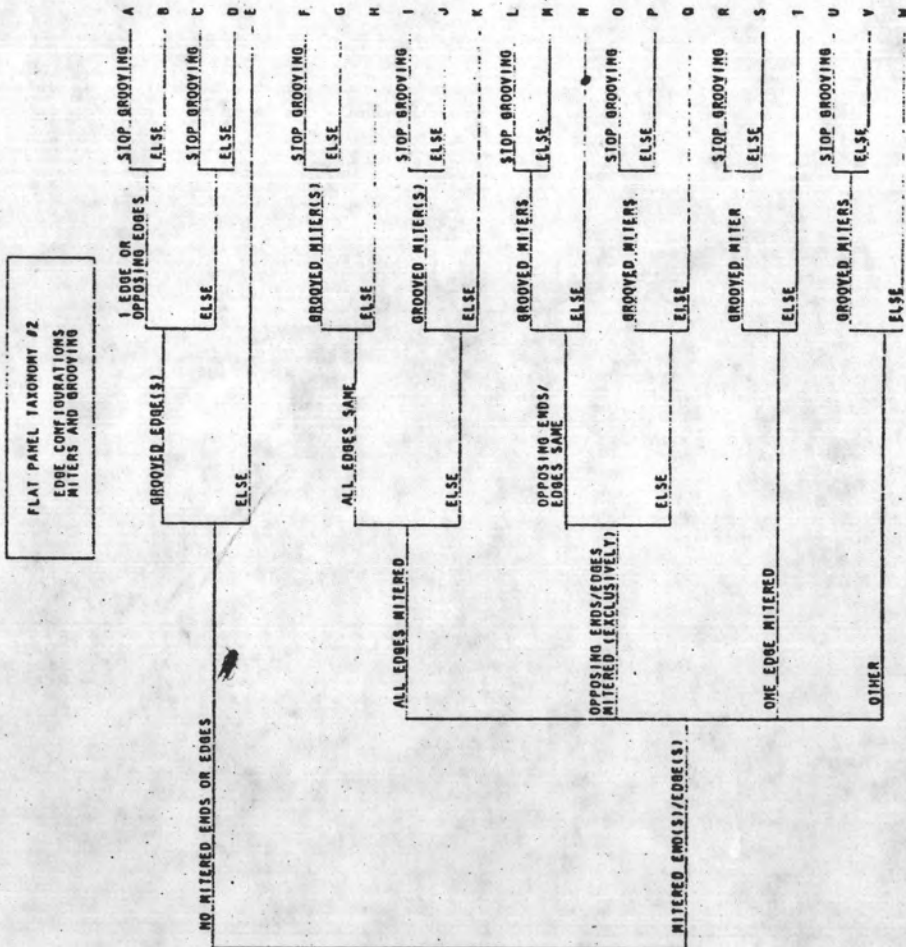
A



B

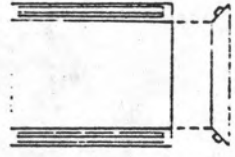
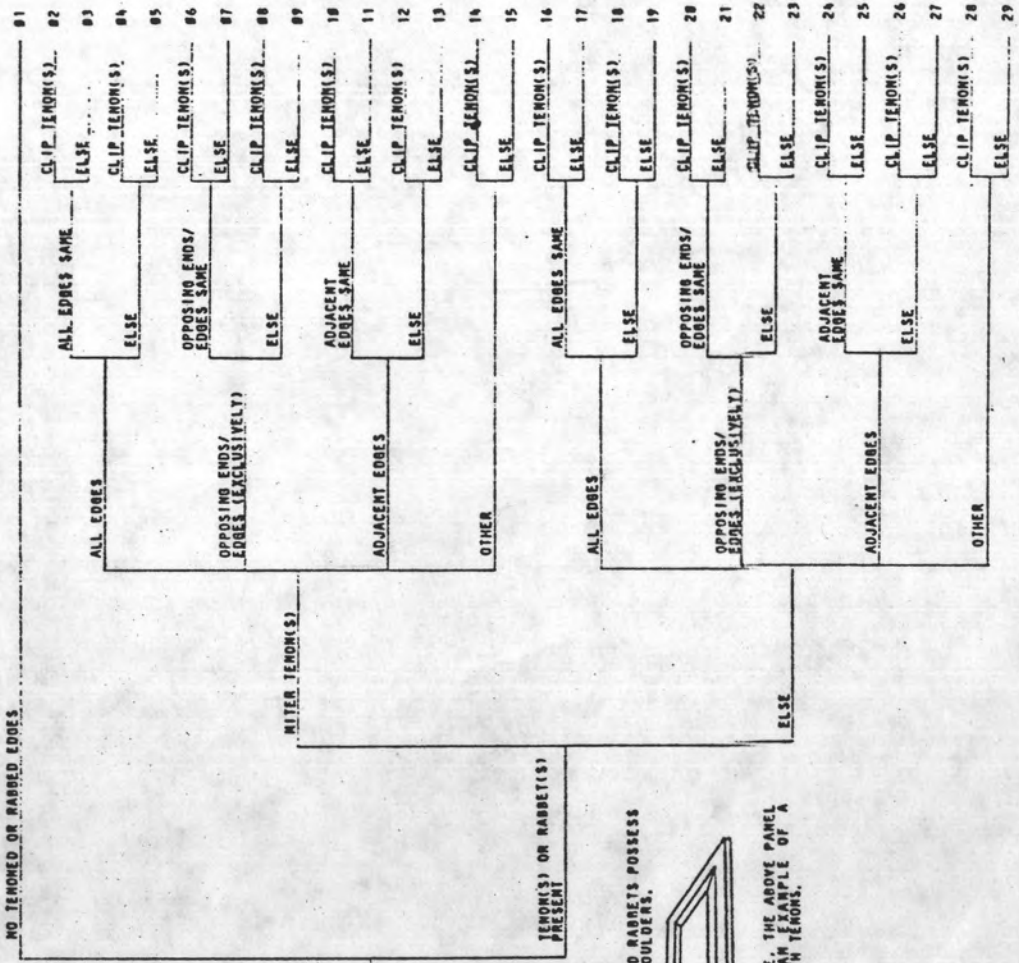


C



5  
FPMMA

FLAT PANEL TAXONOMY #3  
EDGE CONFIGURATIONS  
TENONS AND RABBETS



HEXAGONAL TABLE  
SIDE PANEL  
#6



DOOR CENTER  
PANEL  
#7



END PANEL  
#22

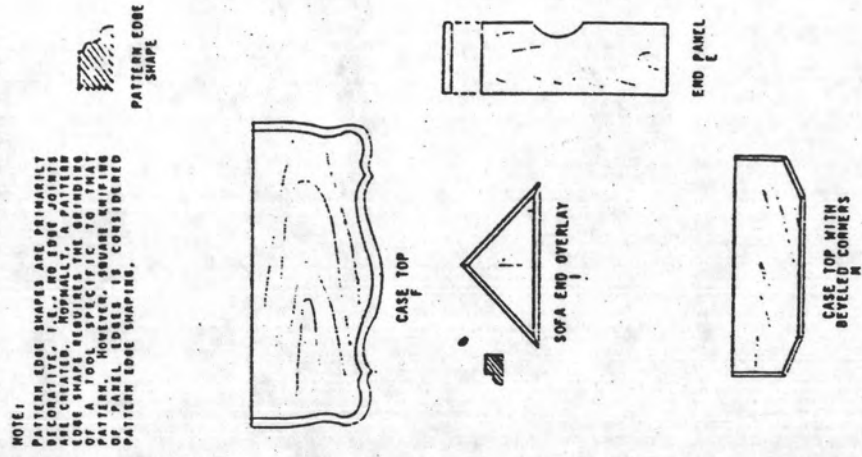
67  
FPMHANN

NOTE: TENONS AND RABBETS POSSESS  
SQUARE SHOULDERS.



THEREFORE, THE ABOVE PANEL  
IS NOT AN EXAMPLE OF A  
PANEL WITH TENONS.

FLAT PANEL TAXONOMY IN  
EDGE CONFIGURATION  
PATTERN SHAPES

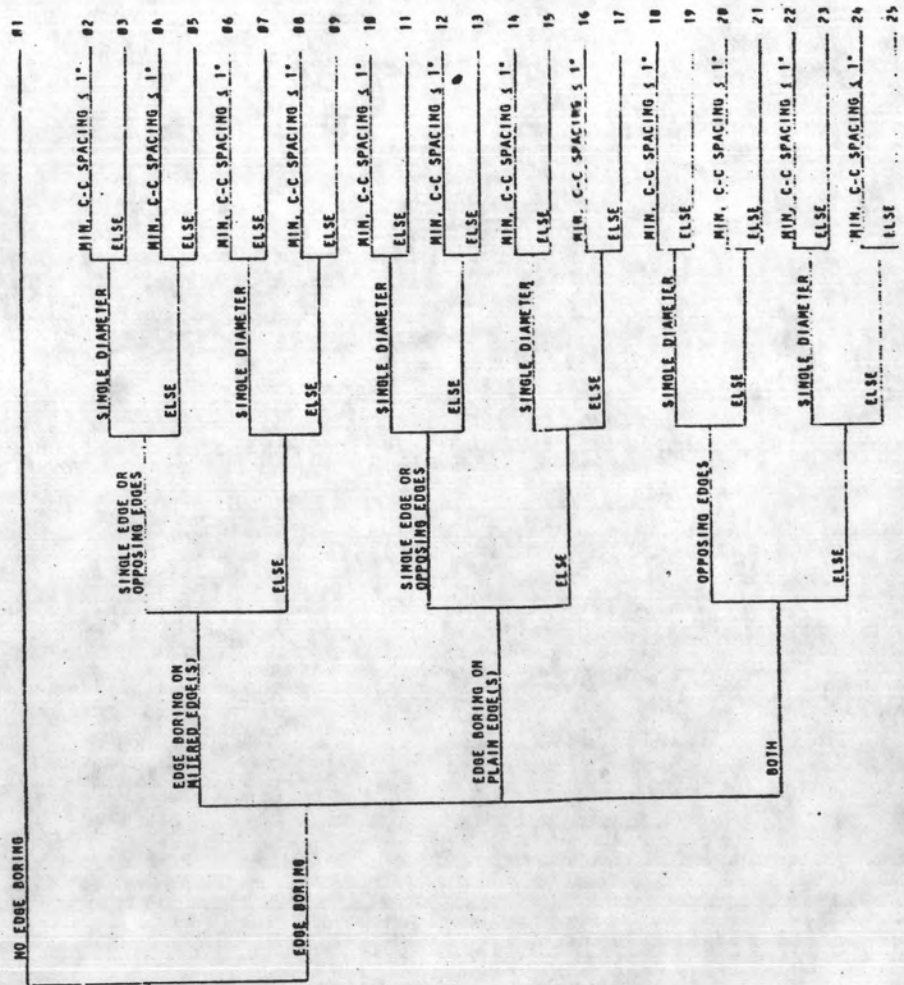


NOTE:  
PATTERN EDGE SHAPES ARE PRIMARILY  
DECORATIVE, I.E., NO EDGE JOINTS  
ARE CREATED. NORMALLY, A PATTERN  
EDGE SHAPE REQUIRES THE BRINDING  
TO BE TO THE INSIDE OF THE  
PATTERN. HOWEVER, SQUARE TRIMMING  
OF PANEL EDGES IS CONSIDERED  
PATTERN EDGE SHAPING.



PPHAWHA

FLAT PANEL TAXONOMY #5  
EDGE/END BORING INFORMATION

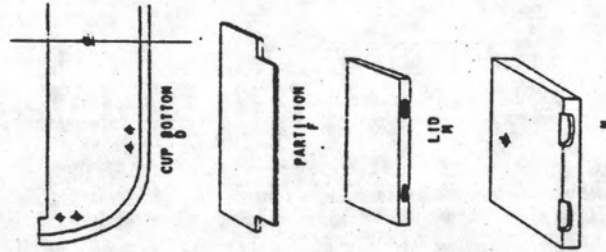
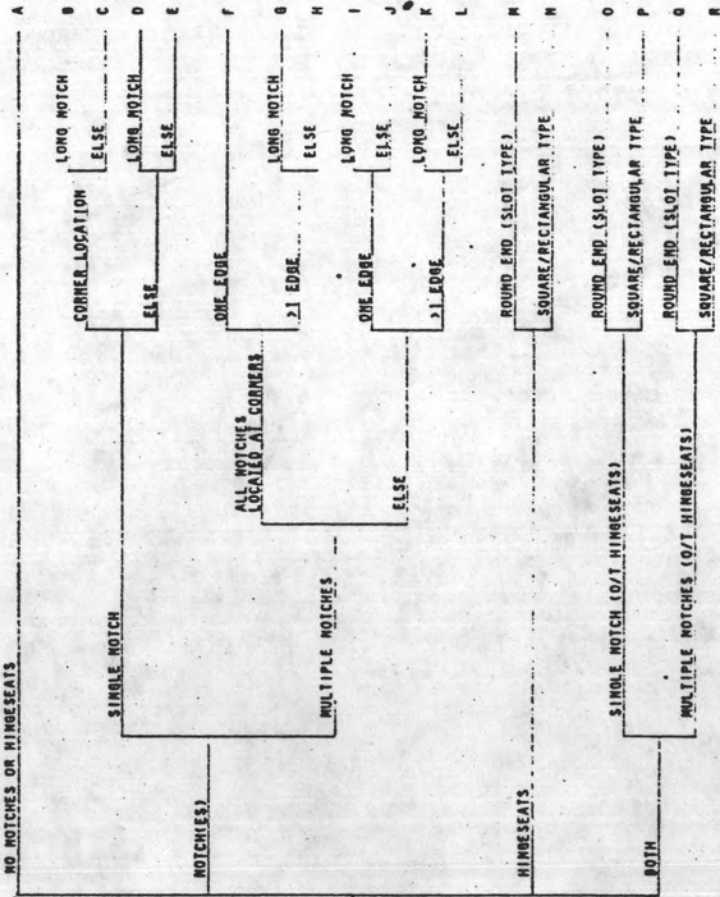


9 10  
V  
FPMANNANN



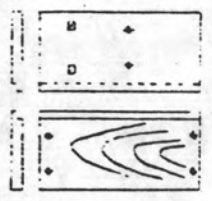
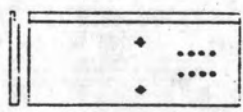
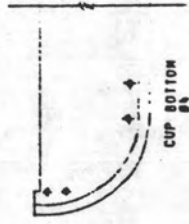
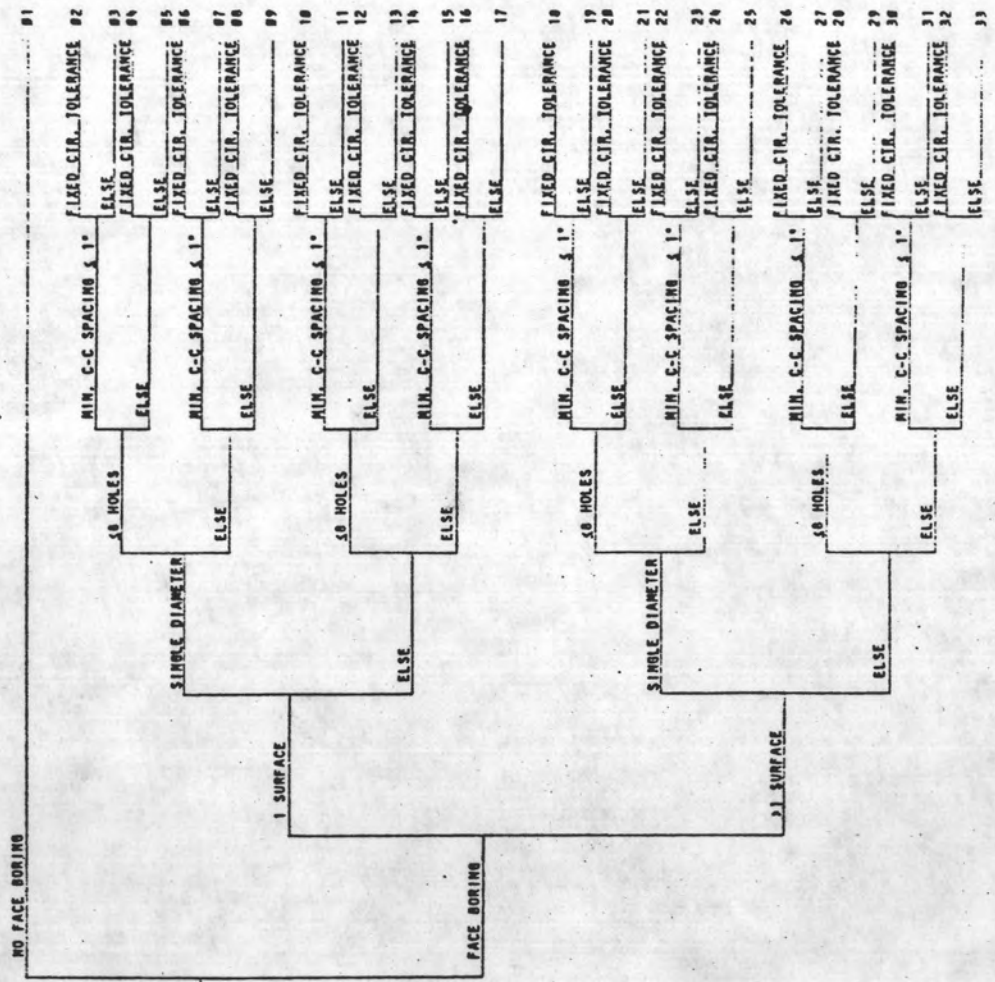
FLAT PANEL TAXONOMY #6  
EDGE CONFIGURATIONS  
NOTCHES & HINGESEATS

NOTE: A LONG NOTCH IS A NOTCH WITH A LENGTH THAT COVERS AT LEAST HALF OF THE EDGE ON WHICH IT IS LOCATED.



11  
PPMARRA

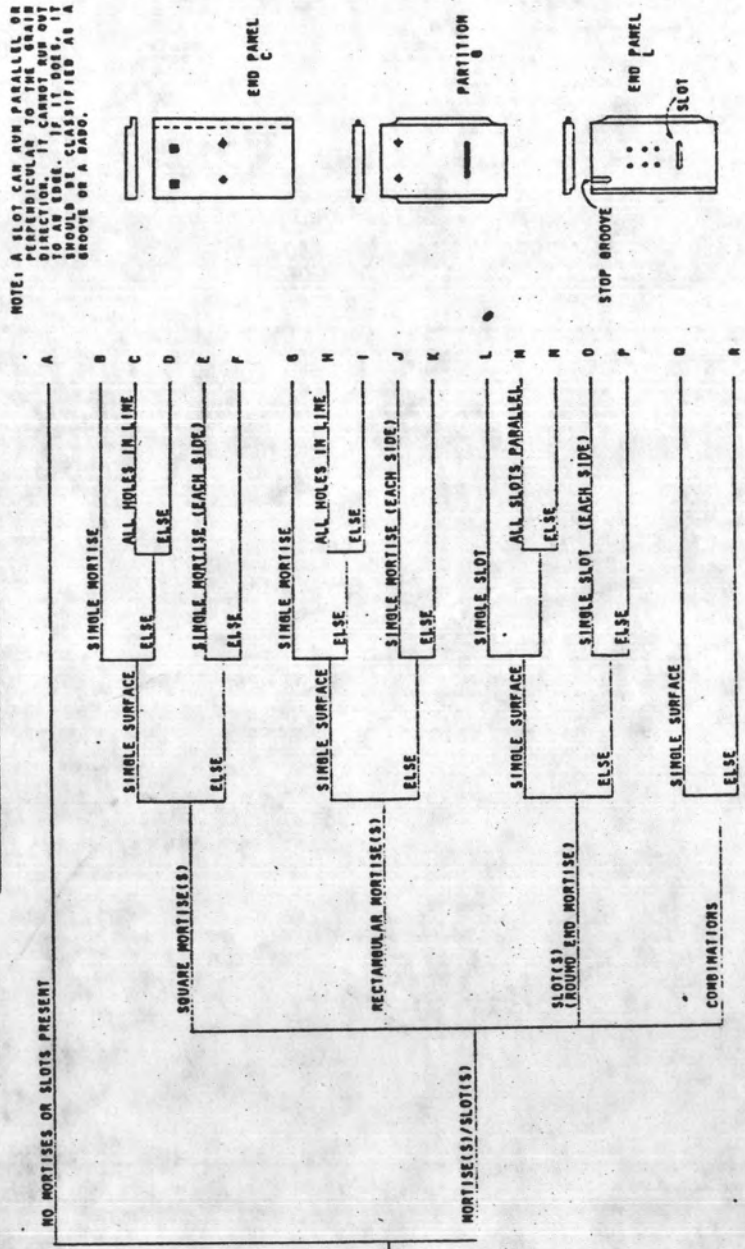
FLAT PANEL TAXONOMY #7  
FACE BORING INFORMATION



12 13  
PPHAWHAWHAW



FLAT PANEL TAXONOMY #6  
SURFACE FEATURES  
MORTISES AND SLOTS

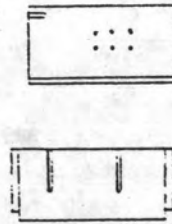
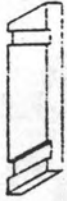


17  
FPMANMANNA

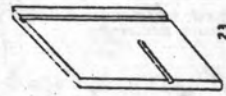
FLAT PANEL TAXONOMY #9  
SURFACE FEATURES  
DADOES AND GROOVES

NOTE: DADOES AND GROOVES

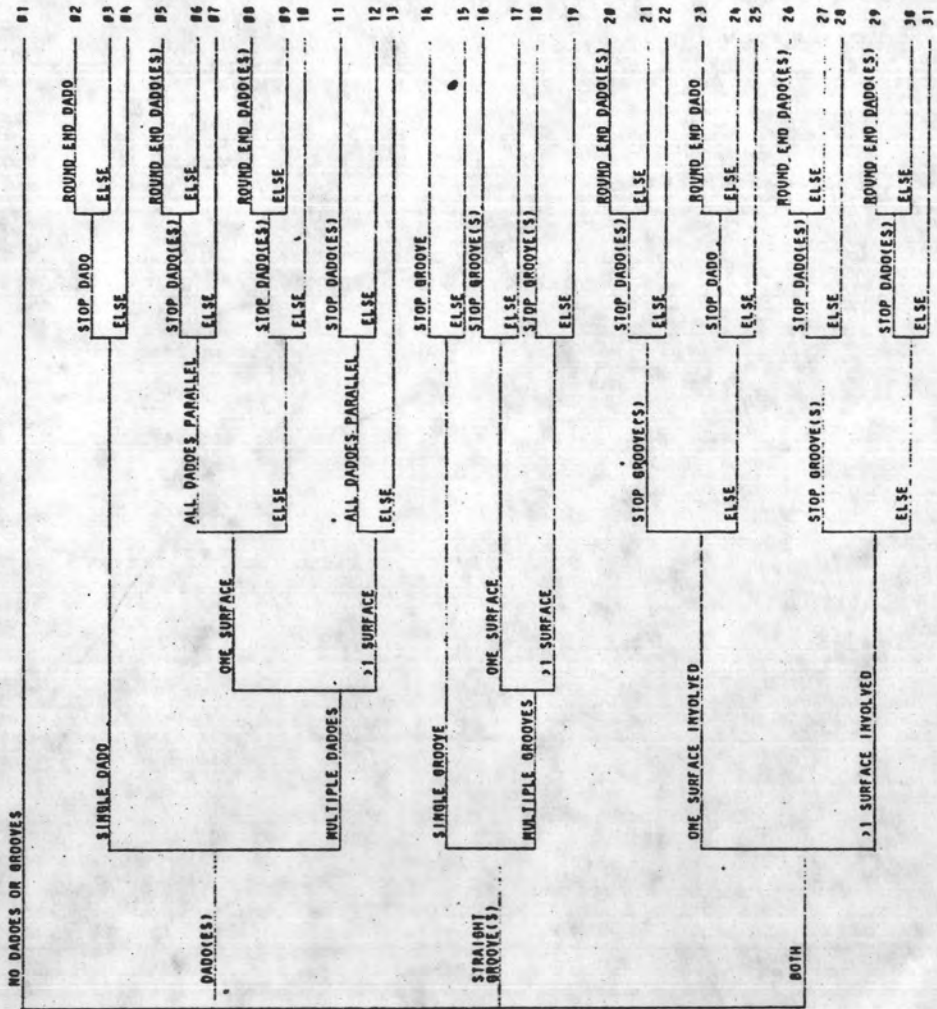
1. A DADO MUST RUN IN A CROSS SECTION OF THE PART PERPENDICULAR TO THE GRAIN OR DIAGONALLY ACROSS IT.
2. A GROOVE MUST RUN PARALLEL TO THE GRAIN OF THE PART. THEREFORE, ON ANY SINGLE PART, ALL GROOVES MUST RUN PARALLEL TO EACH OTHER.



14

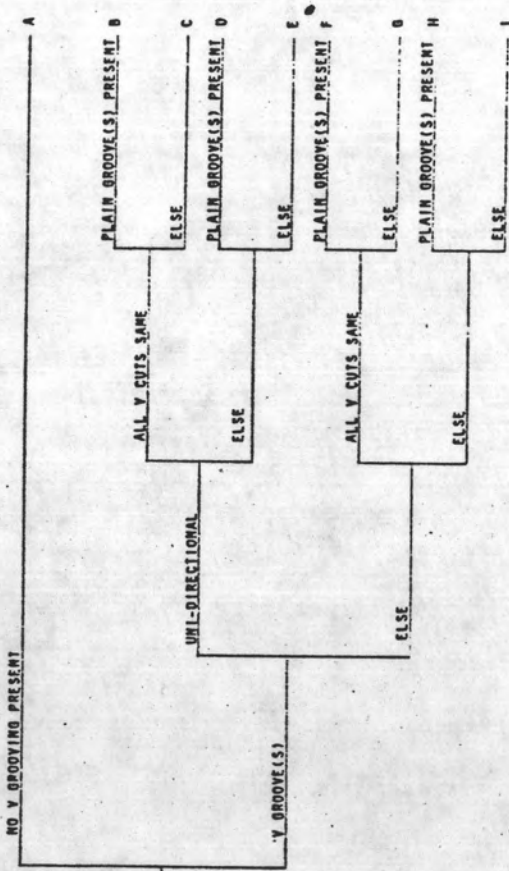
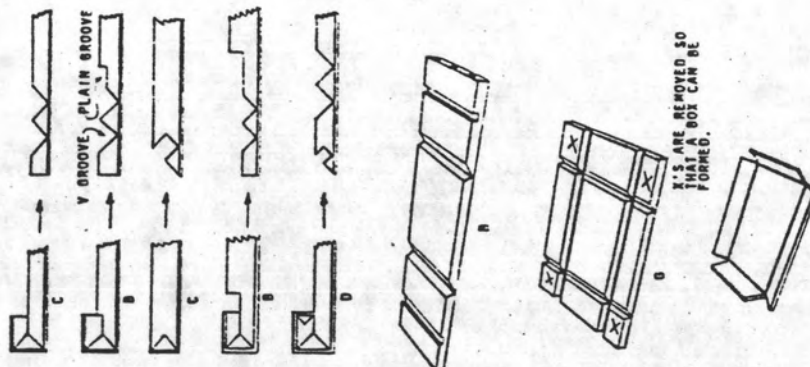


23



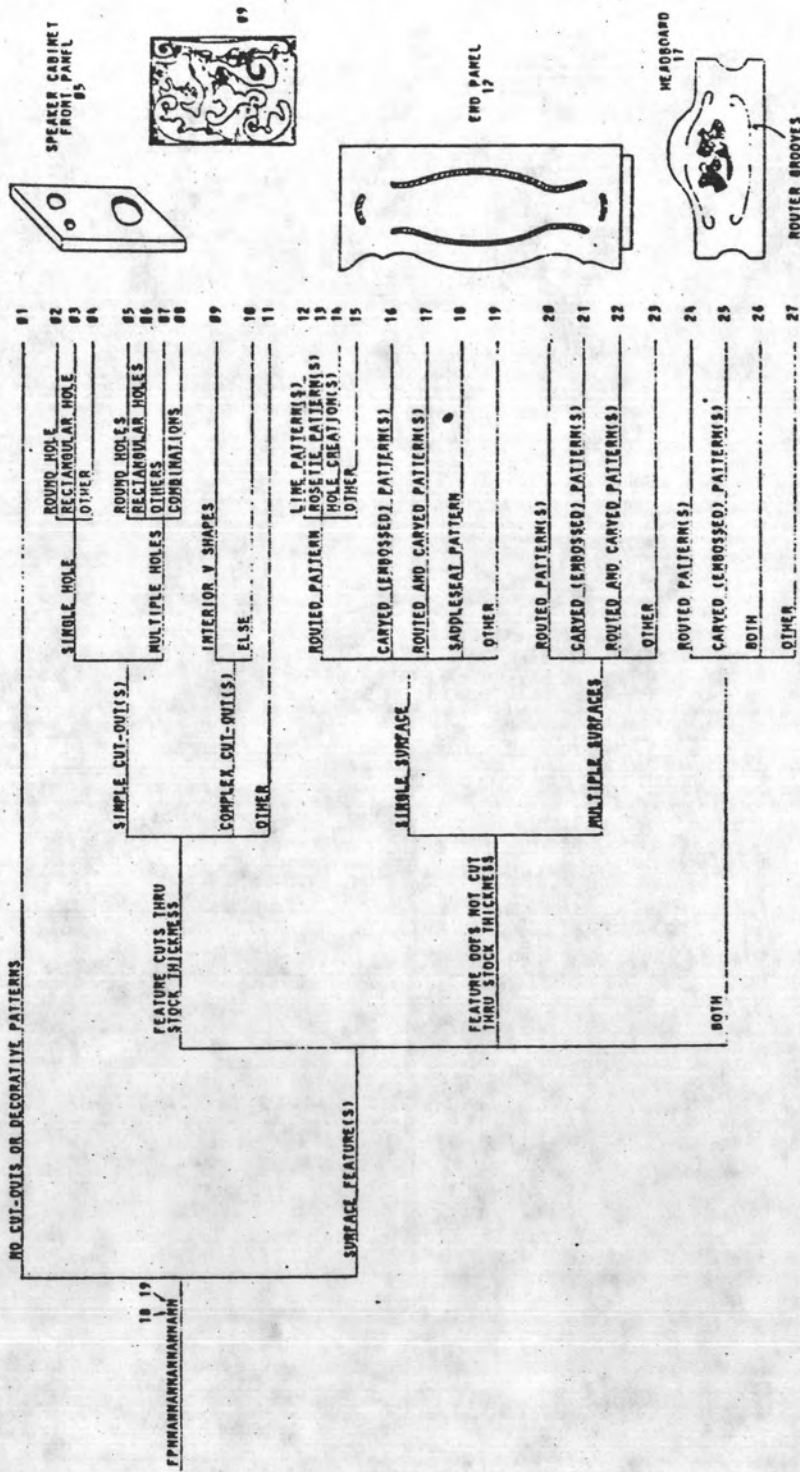
15 16  
FPMANMANMAN

FLAT PANEL TAXONOMY #18  
SURFACE FEATURES  
V GROOVING



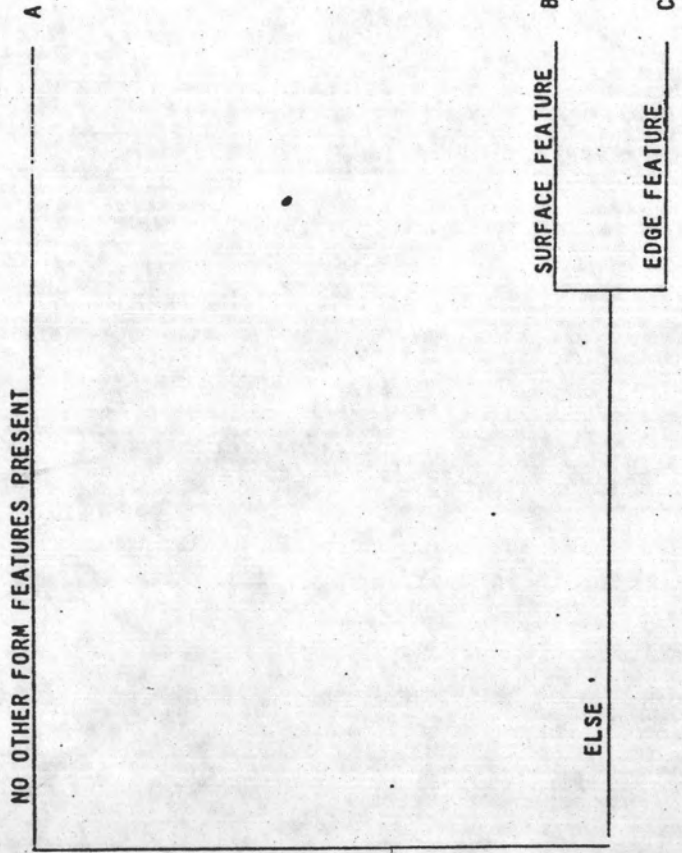
17  
PPHANNANNAHNA

FLAT PANEL TAXONOMY #11  
SURFACE FEATURES  
CUT-OUTS AND DECORATIVE PATTERNS



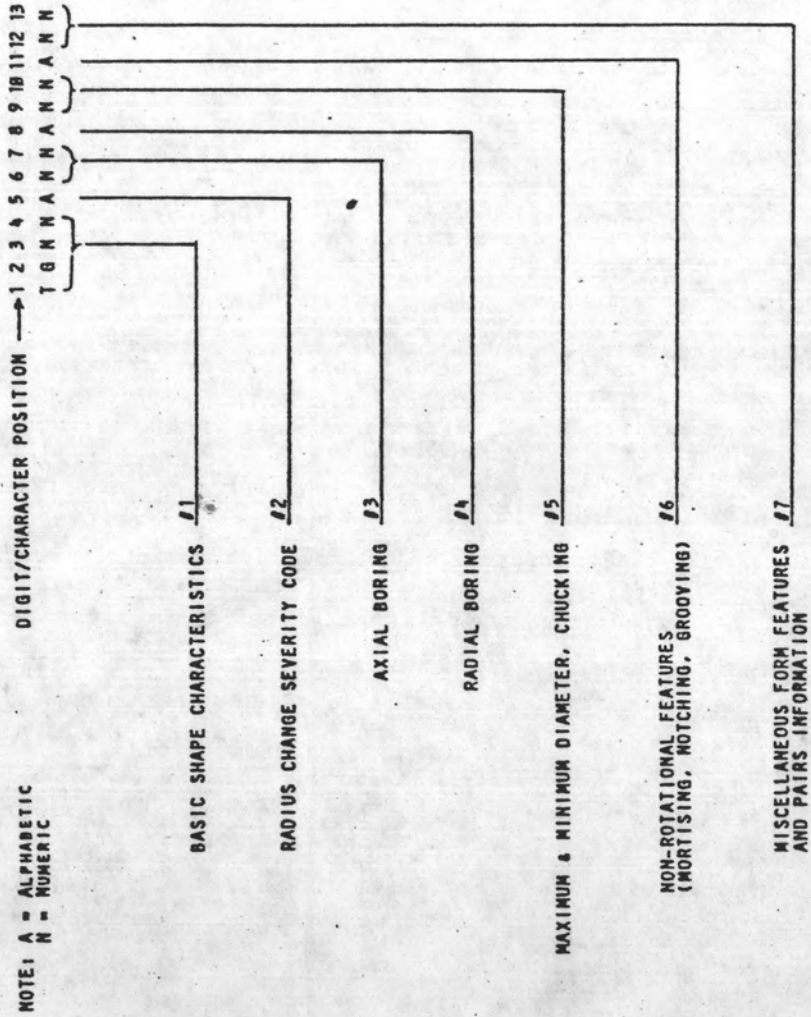
10 19  
FPHANNANNANNANNANN

FLAT PANEL TAXONOMY #12  
OTHER FORM FEATURES



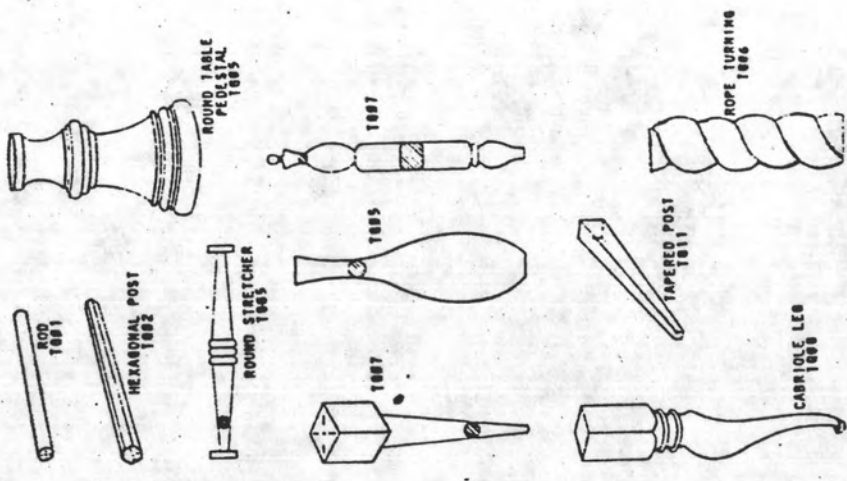
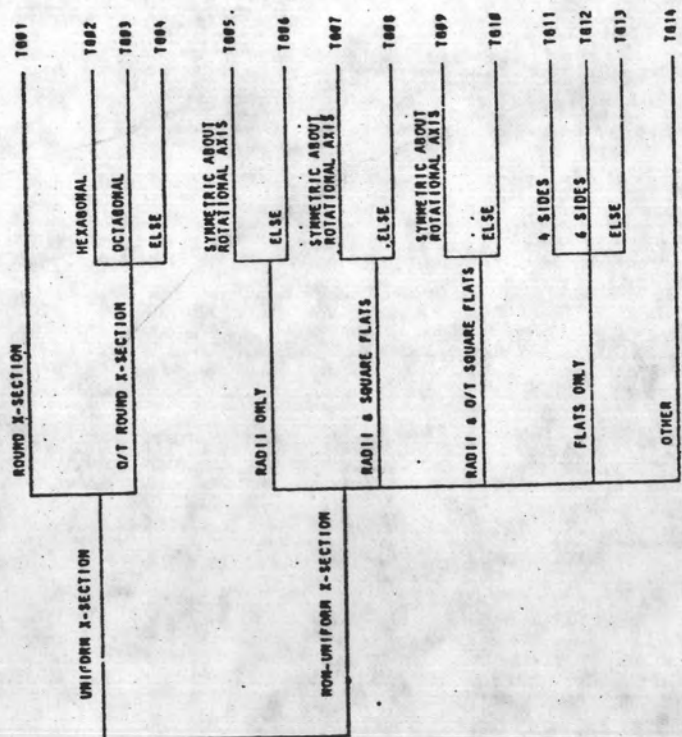
20  
↓  
FPNNANNANNANNANNA

TURNING TAXONOMIES

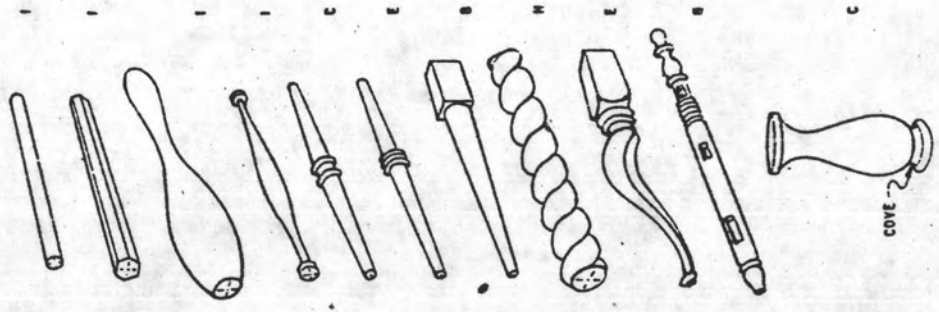


TURNING TAXONOMY #1  
BASIC SHAPE CHARACTERISTICS

NOTE: TURNINGS INCLUDE ALL PARTS THAT CAN BE MANUFACTURED IN A ROTATIONAL PROCESS. ROTATIONAL PROCESSING IS ASSOCIATED WITH LATHE, TURNING AND BACKDRIFT, COPY LATHES, MILLING MACHINES, AND MULTIPLE SPINDLE TURNING MACHINES. PARTS CAN BE MADE ON TURNING MACHINES. IT IS ARBITRARY AS TO WHETHER THEY ARE CLASSIFIED AS STRAIGHT RAILS OR TURNINGS. THE TAXONOMY WILL ACCOMMODATE THEM EITHER WAY.



**TURNING TAXONOMY #2**  
**RADIUS CHANGE INFORMATION**



A	RCSC = (1)
B	RCSC = (2)
C	RCSC = (3)
D	RCSC = (1) & (2)
E	RCSC = (1) & (3)
F	RCSC = (2) & (3)
G	RCSC = (1), (2), (3)
H	RCSC = (4)
I	ELSE

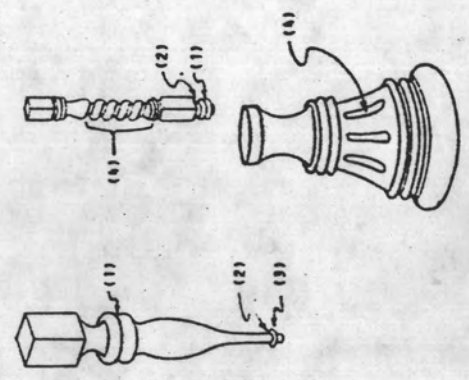
1  
TOWNA

NOTE: RCSC = (4)  
IF A TURNING HAS NON-ROTATIONAL FORM FEATURES WHICH DO NOT FIT ANY OF THE PATTERNS, IT WILL BE CLASSIFIED AS A DISTINCT CATEGORY OF PARTS. NO COMBINATIONS OF RCSC #s WITH OTHER RCSC VALUES WILL BE NECESSARY.

**RADIUS CHANGE SEVERITY CODE - RCSC**  
THE RCSC IS DESIGNED TO REFLECT THE DEGREE OF THE RADIUS CHANGE. THE RADIUS OF THE PARTS, THEREFORE, SHOULD BE INDICATED FOR PARTS SUCH AS BLOCKS AND CHUCKED ENDS, SHOULD BE IGNORED WHILE ASSIGNING THE RCSC.

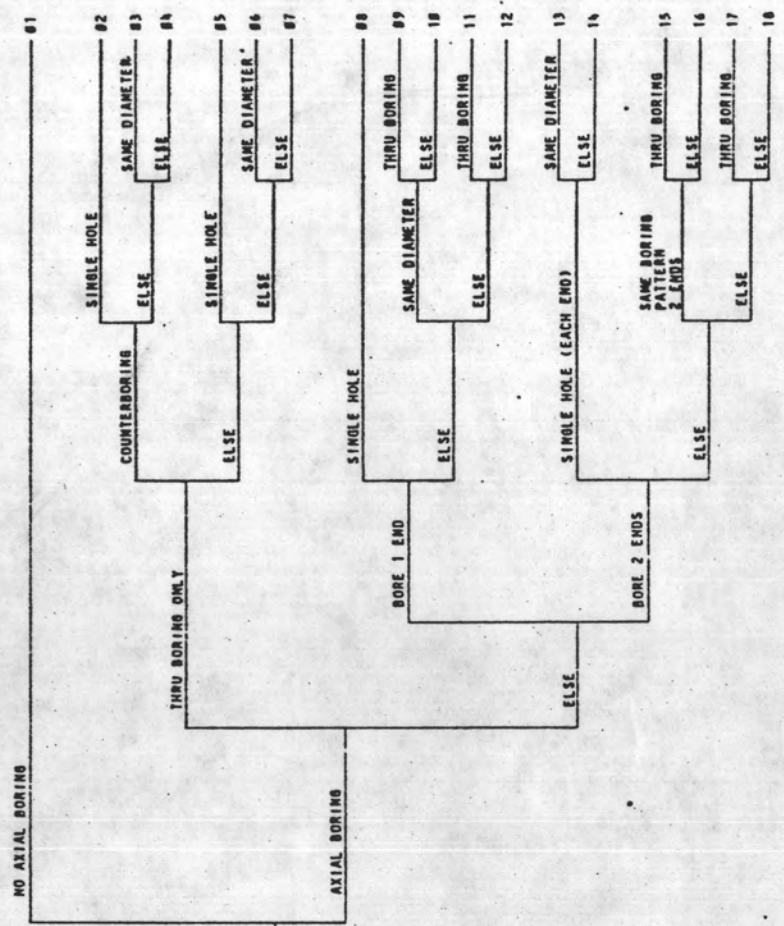
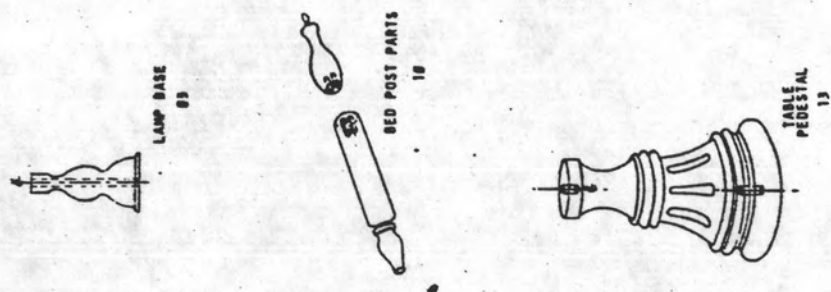
- CHARACTERISTIC**
- (1) INTERIOR V SHAPES
  - (2) ABRUPT CHANGES IN PART RADIUS (DISTINCT STEPS  $\geq 1/4$ )
  - (3) BEADS AND/OR COVES
  - (4) NON-ROTATIONAL FEATURES (FLUTES, CARVING, ETC.)

**EXAMPLES**



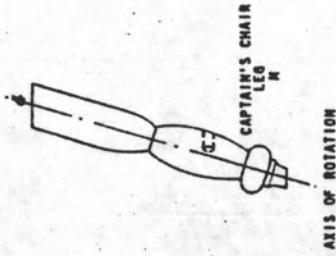
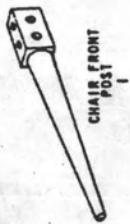
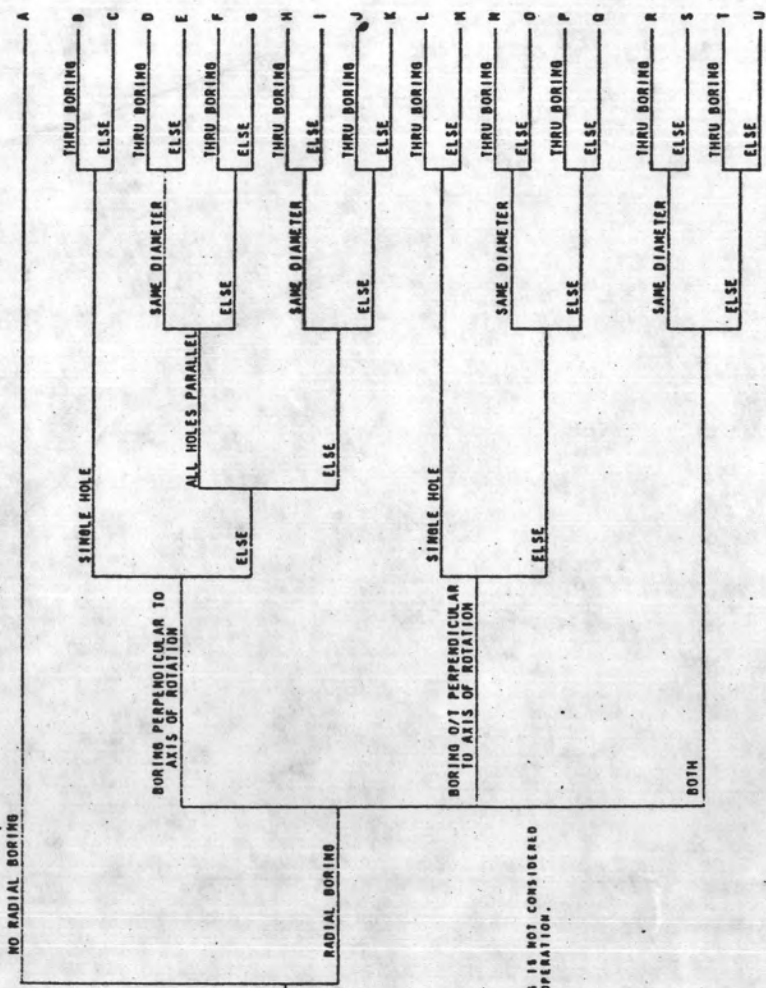


TURNING TAXONOMY #5  
AXIAL BORING INFORMATION



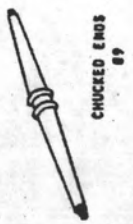
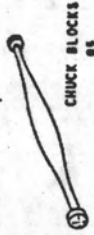
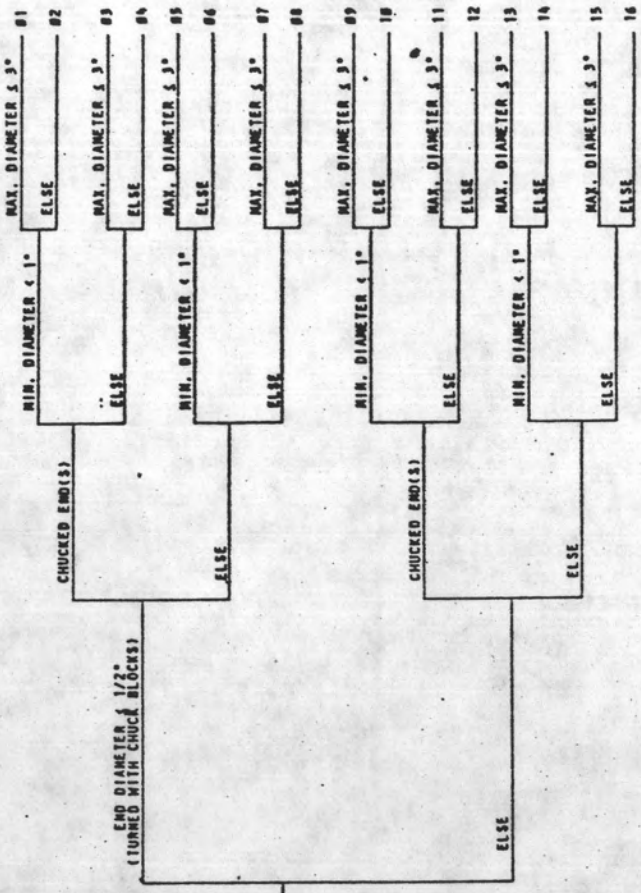
6 /  
TOMMANK

TURNING TAXICOPY PA  
RADIAL BORING INFORMATION



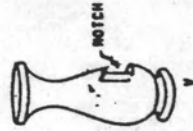
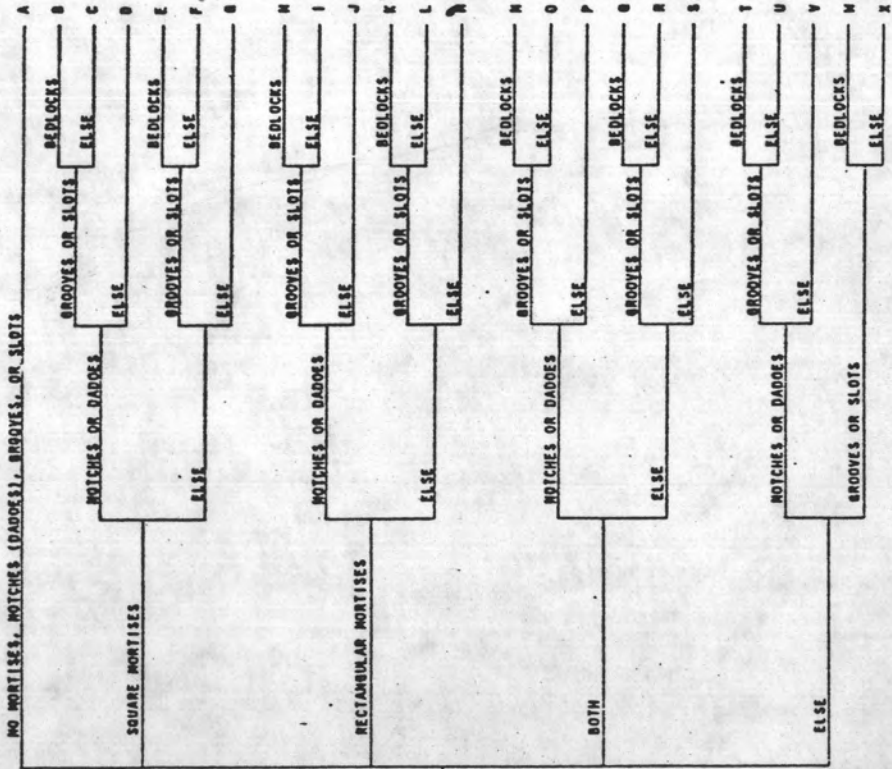
NOTE: BEDLOCKING IS NOT CONSIDERED  
A BORING OPERATION.

TURNING TAXONOMY #5  
MAX.-MIN. DIAMETERS, CHUCKING



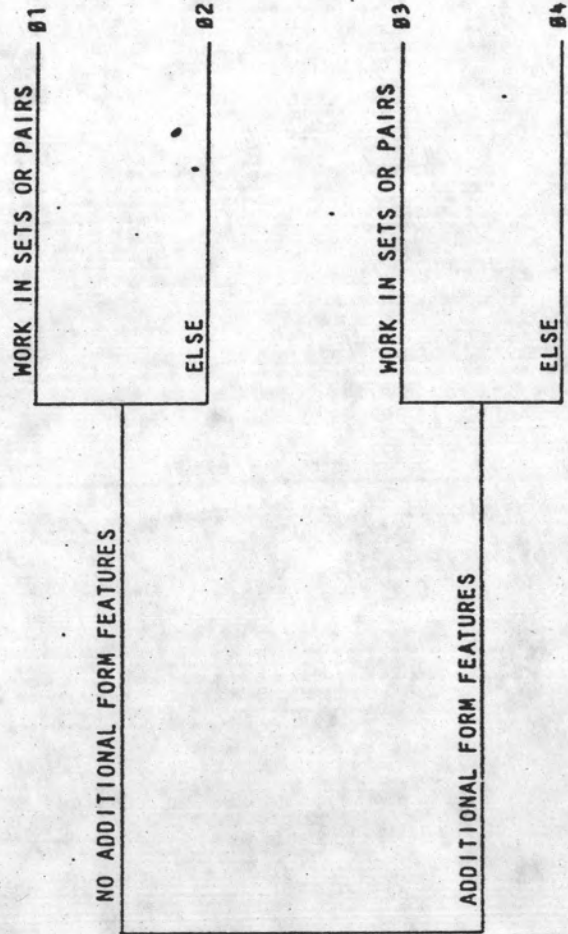
1/2  
TORNHAWK

TURNING TAXONOMY #6  
NON-ROTATIONAL FEATURES  
(MORTISING, NOTCHING, GROOVING)



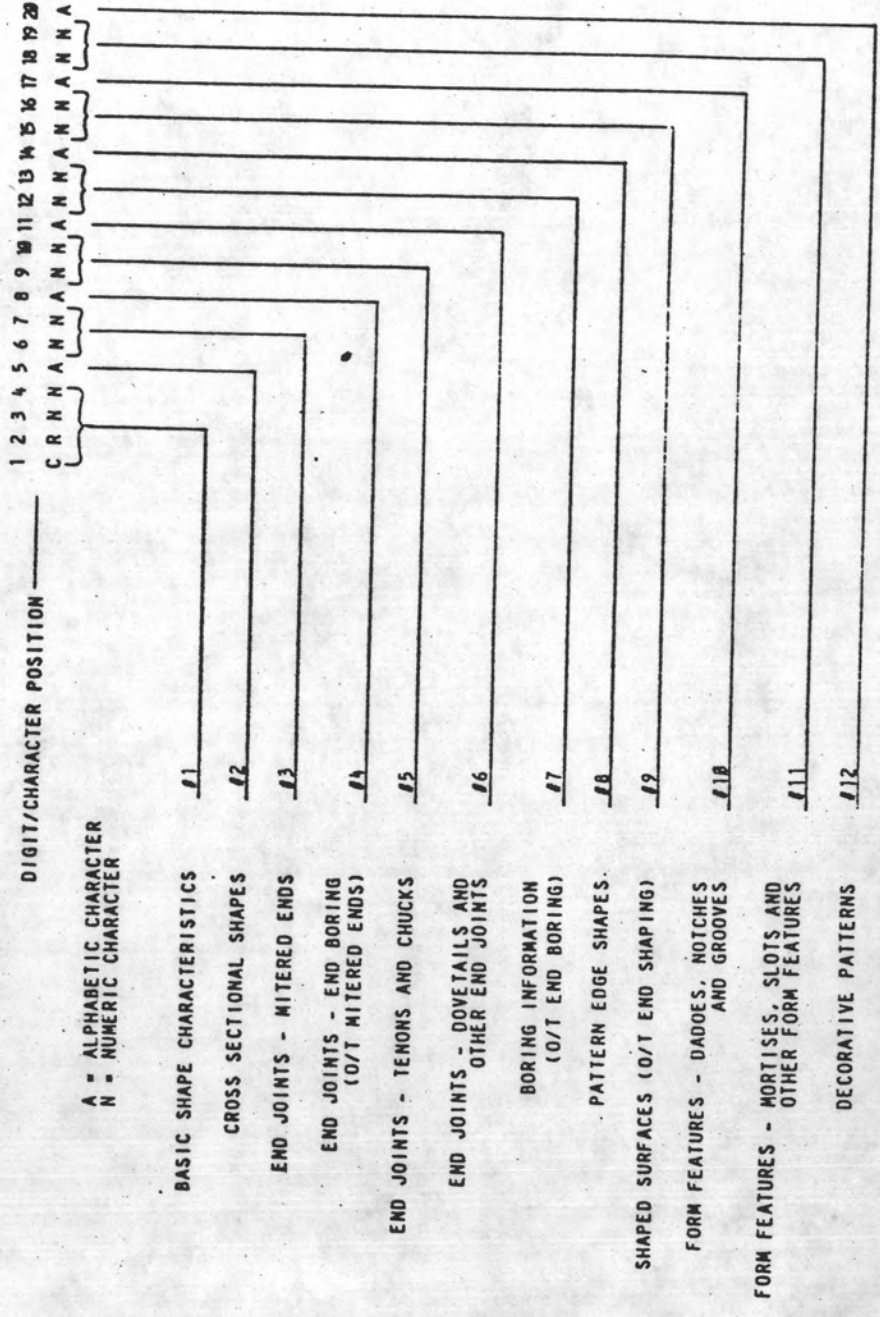
11  
TENNANNA

TURNING TAXONOMY #7  
MISCELLANEOUS FORM FEATURES  
AND PAIRS INFORMATION



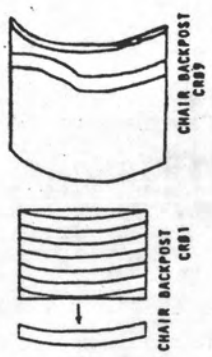
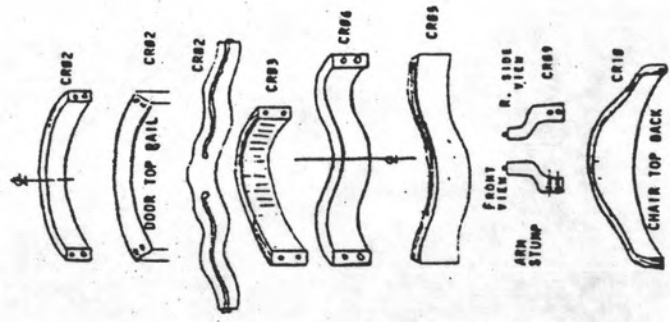
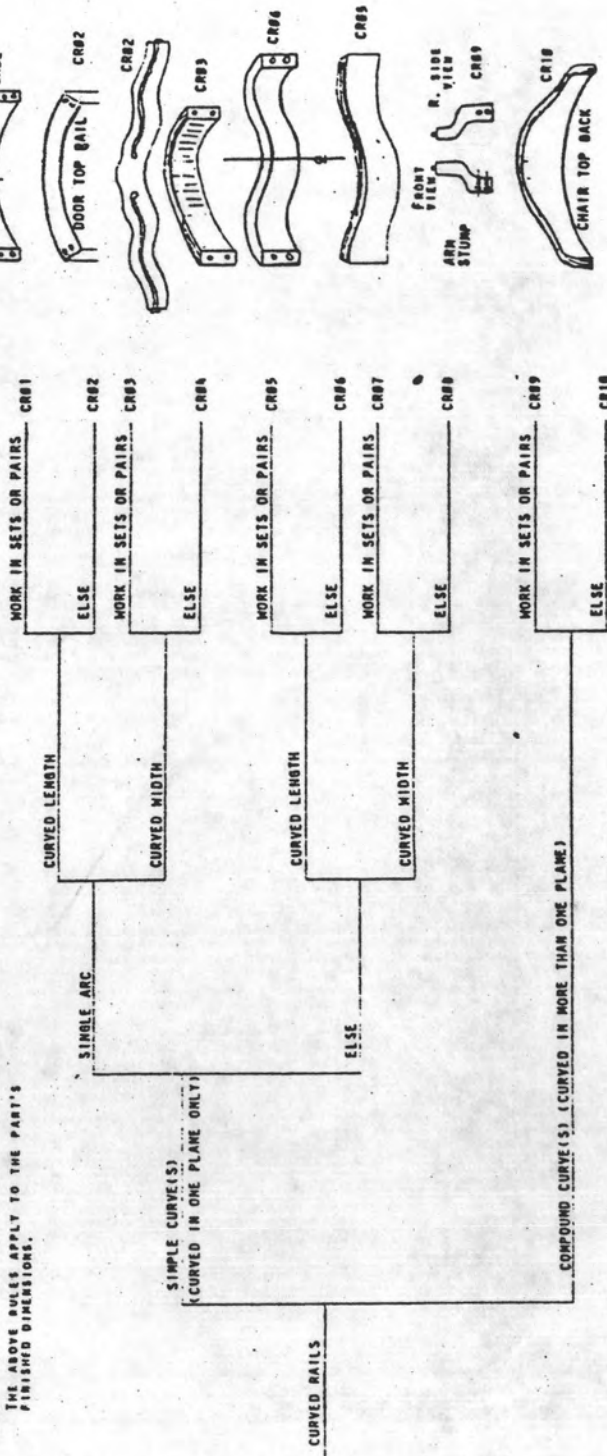
12 13  
TGNHANNANNH

CURVED RAIL TAXONOMIES



**CURVED RAIL TAXONOMY #1  
BASIC SHAPE CHARACTERISTICS**

NOTE: A CURVED RAIL IS A PART SUCH THAT:  
 1. THE WIDTH  $\neq 0$ .  
 2. NO STRAIGHT AXIS EXISTS THRU ITS LENGTH OR THERE ARE NO STRAIGHT FLAT EDGES ON THE PART.  
 THE ABOVE RULES APPLY TO THE PART'S FINISHED DIMENSIONS.



NOTE: THIS IS NOT A CURVED RAIL SINCE A STRAIGHT AXIS EXISTS THRU THE PART.

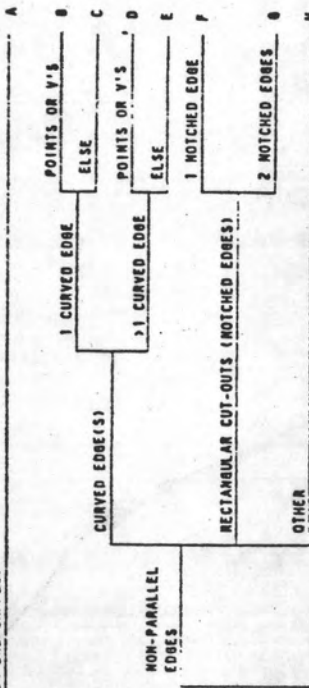
NOTE: BERT PARTS CONSTITUTE A LARGE PORTION OF THE CURVED RAIL POPULATION.  
 TYPICAL PARTS  
 TABLE RIMS (APRONS)  
 FRAME DOOR TOP RAILS  
 BASE FRAME RAILS  
 CHAIR BACKPOSTS  
 CHAIR STRETCHERS  
 CHAIR RAILS & SIDE RAILS  
 ARM BOWS (CAPTAIN'S CHAIRS)

NOTE: THIS IS NOT A CURVED RAIL SINCE A STRAIGHT AXIS EXISTS THRU THE PART.

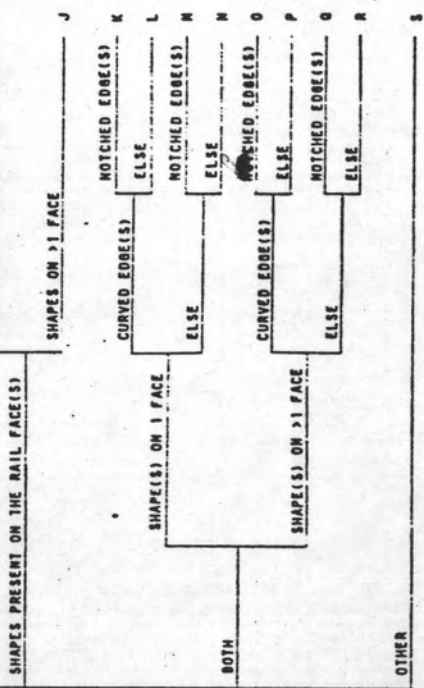
CURVED RAIL TAXONOMY #2  
X-SECTIONAL SHAPE

NOTE: UNIFORMITY OF CROSS SECTIONAL SHAPE AND FINISH THROUGH THE ENTIRE PART CONSTITUTION. UNIFORMITY IS NOT DISTURBED BY END JOINTS OR SHAPES OR BY FORM FEATURES, SUCH AS BORED HOLES, NOTCHES, ETC.

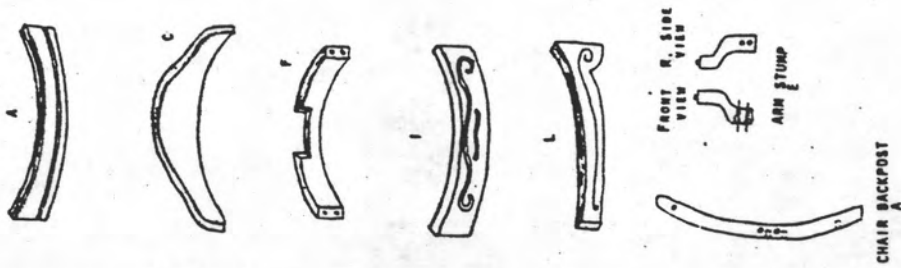
UNIFORM X-SECTION (O/T END JOINTS OR END SHAPES)



NON-UNIFORM X-SECTION



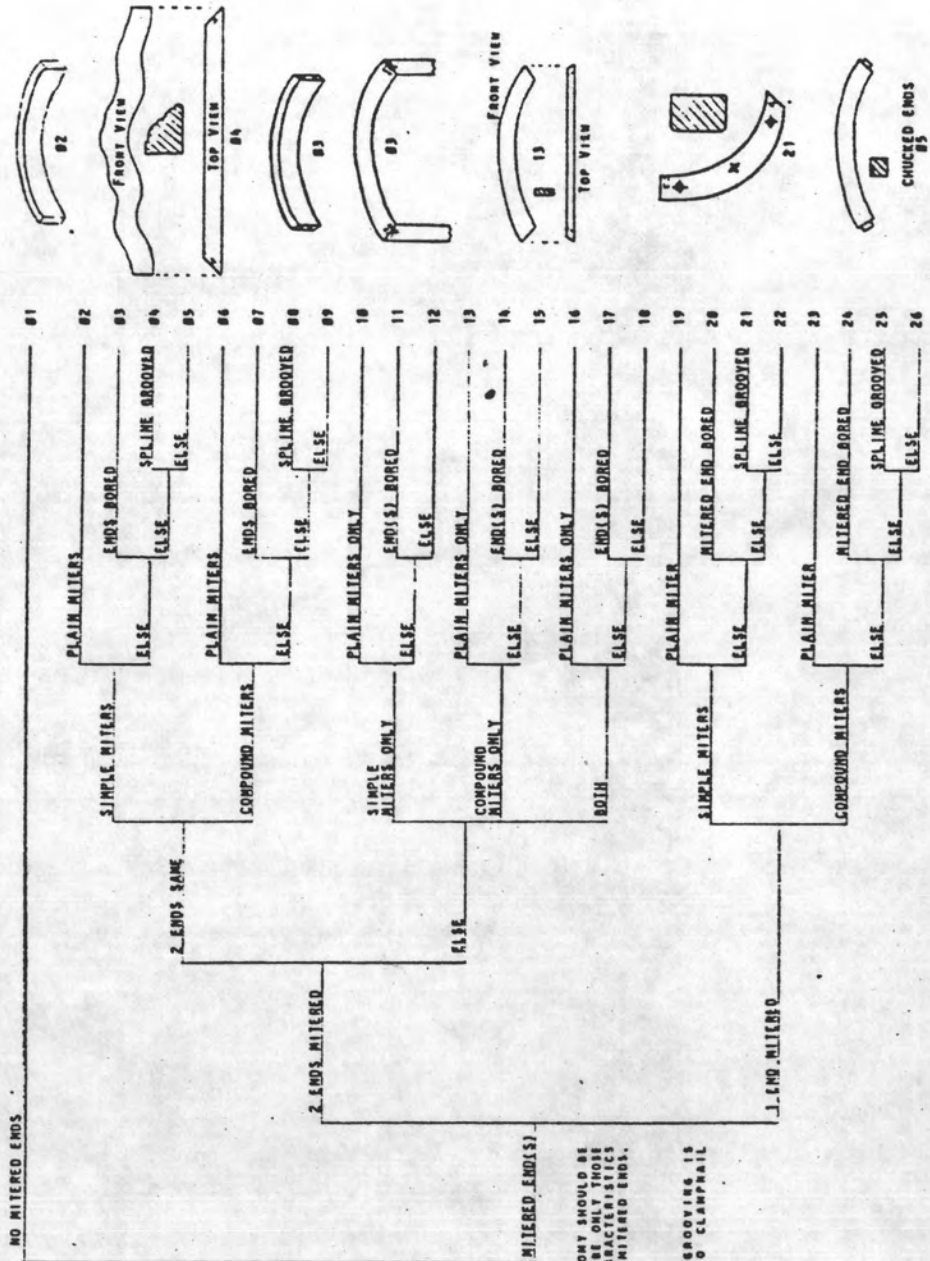
NOTE: A CURVED RAIL NORMALLY HAS 2 EDGES, WITH THE THICKNESS DIMENSION CONSTITUTING AN EDGE. THIS IS TRUE FOR ANY RAIL WHERE W/T > 1. FOR CURVED RAILS WITH A SQUARE CROSS SECTION (W/T = 1), A TOGGED CRUISE.



CRMA



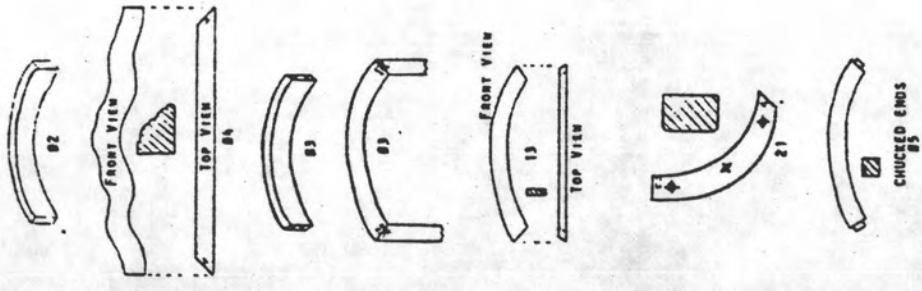
CURVED RAIL TAXONOMY #3  
END JOINTS - MITERED ENDS



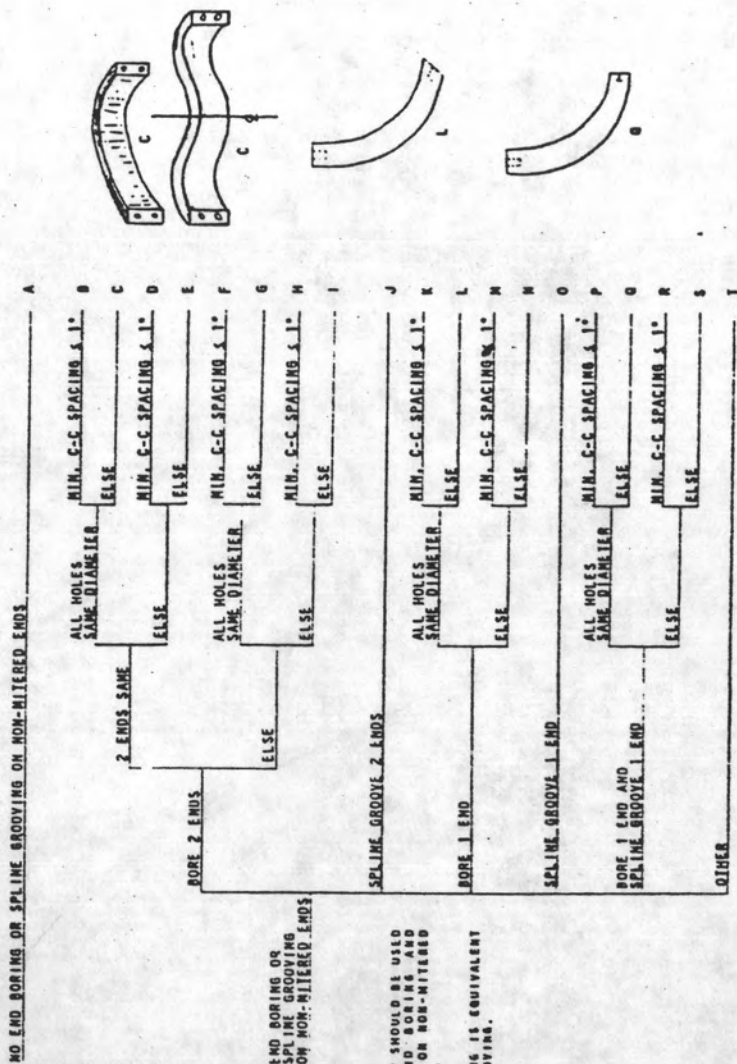
67  
CERNANN

NOTE: THIS TAXONOMY SHOULD BE USED TO DESCRIBE ONLY THOSE JOINT CHARACTERISTICS THAT OCCUR ON MITERED ENDS.

NOTE: SPLINE GROOVING IS EQUIVALENT TO CLAMP RAIL GROOVING.



CURVED RAIL TAXONOMY #4  
END JOINTS -  
END BORING (O/I MITERED ENDS)

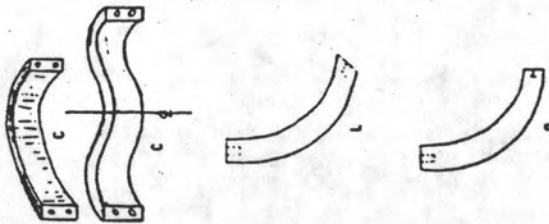


CRHMMMA

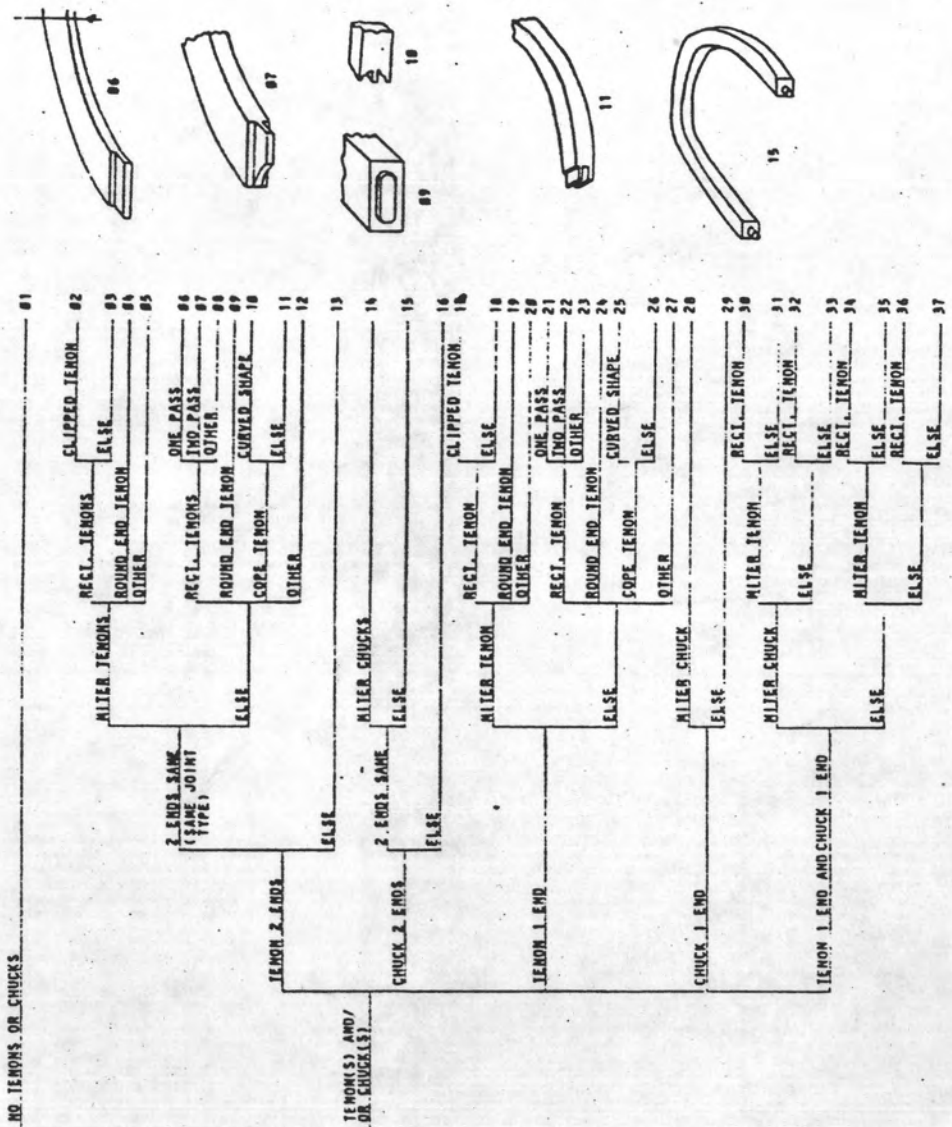
END BORING OR  
SPLINE GROOVING  
ON NON-MITERED ENDS

NOTE: THIS TAXONOMY SHOULD BE USED  
TO DESCRIBE END JOINTS ON  
SPLINE GROOVING OR NON-MITERED  
ENDS EXCLUSIVELY.

NOTE: SPLINE GROOVING IS EQUIVALENT  
TO CLAMPTRAIL GROOVING.

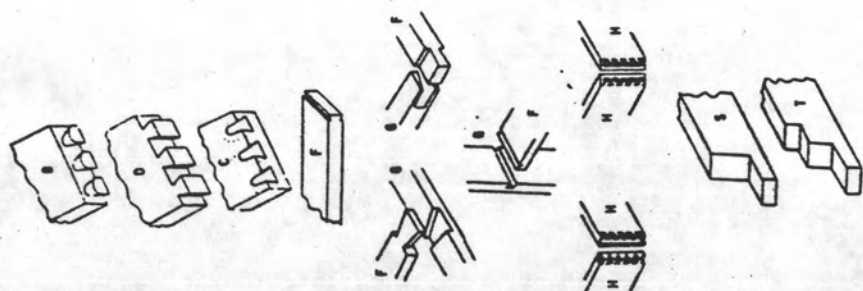
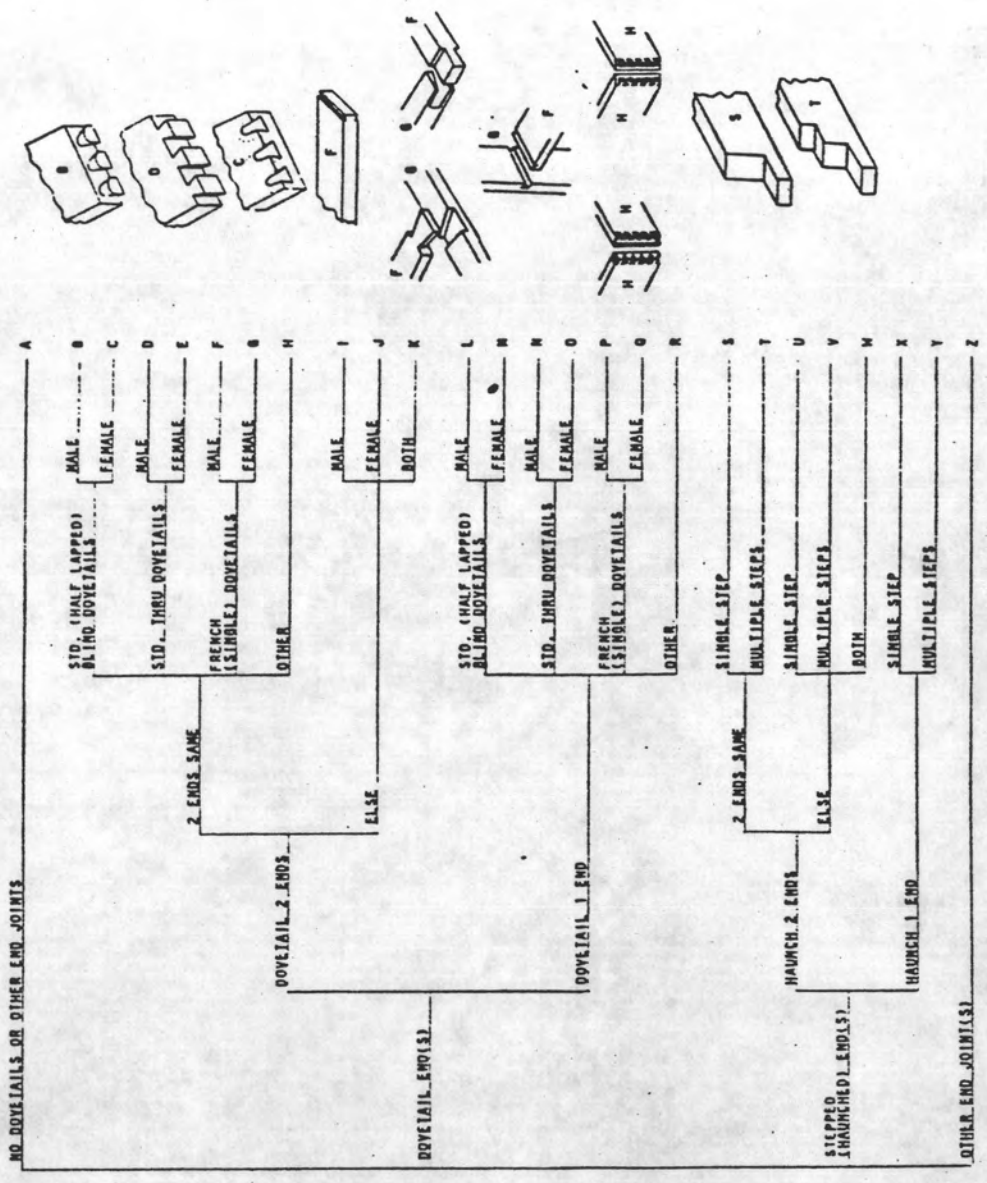


CURVED RAIL TAXONOMY #5  
END JOINTS -  
TENONED AND CHUCKED ENDS



9 10  
CAMBODIA

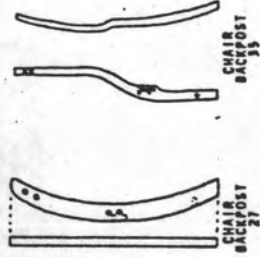
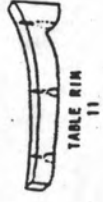
CURVED RAIL TAXONOMY #6  
 END JOINTS -  
 DOVETAILS AND OTHER END JOINTS



11  
 CRIMMANNIA

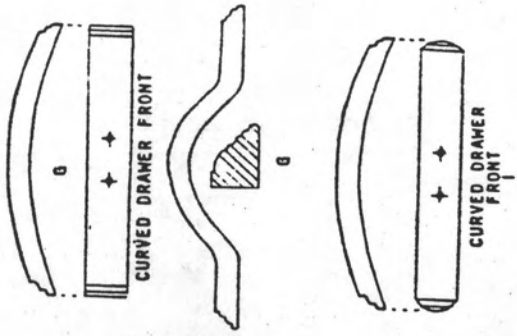
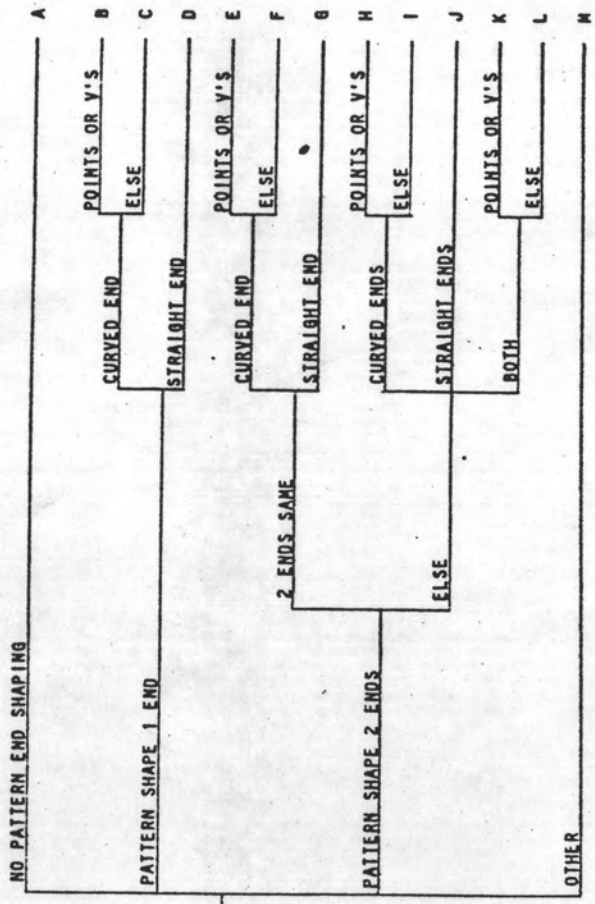
CURVED RAIL TOLERANCE 87  
BORING INFORMATION - O/T END BORING

NO BORING(O/T END BORING)		81					
SIMPLE CURVED SURFACE	ALL HOLES SAME DIAMETER	82	MIN. C-C SPACING $\pm 1"$				
	ELSE	83	ELSE				
COMPOUND CURVED SURFACE	84 HOLES	84	MIN. C-C SPACING $\pm 1"$				
	ELSE	85	ELSE				
OTHER	ALL HOLES SAME DIAMETER	86	MIN. C-C SPACING $\pm 1"$				
	ELSE	87	ELSE				
2 SIMPLE CURVED SURFACES	ALL HOLES SAME DIAMETER	88	MIN. C-C SPACING $\pm 1"$				
	ELSE	89	ELSE				
BORE 1 SURFACE	ALL HOLES SAME DIAMETER	90	MIN. C-C SPACING $\pm 1"$				
	ELSE	91	ELSE				
BORE 2 SURFACES	ALL HOLES SAME DIAMETER	92	MIN. C-C SPACING $\pm 1"$				
	ELSE	93	ELSE				
OTHER	ALL HOLES SAME DIAMETER	94	MIN. C-C SPACING $\pm 1"$				
	ELSE	95	ELSE				



12 33  
CRIMMANNANN

CURVED RAIL TAXONOMY #8  
PATTERN END SHAPES

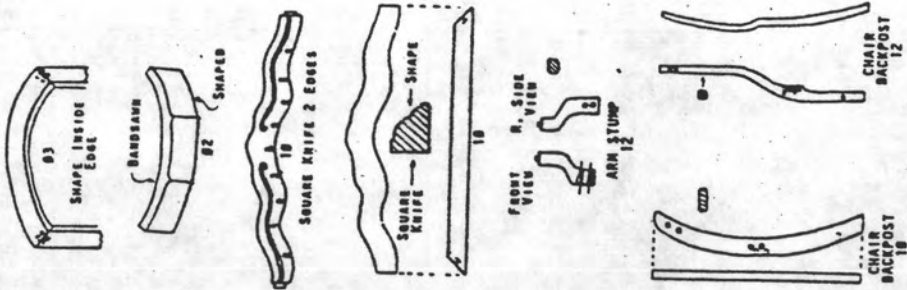


14  
CRNMANNA

NOTE: PATTERN END SHAPES ARE  
PRIMARILY AESTHETIC, I.E.,  
THEY DO NOT INVOLVE JOINTS.

CURVED RAIL TAXONOMY 69  
SHAPED SURFACES O/T END SHAPING

NO. SHAPED SURFACES O/T END SHAPING	
SHAPE 1 SURFACE	81
SINGLE CURVED SURFACE	POINTS OR V'S
ELSE	82
COMPOUND CURVED SURFACE	POINTS OR V'S
ELSE	83
FLAT SURFACE (PRIOR TO SHAPING)	84
OTHER	POINTS OR V'S
ELSE	85
SHAPE 2 SURFACES	86
2 SINGLE CURVED SURFACES	POINTS OR V'S
ELSE	87
2 COMPOUND CURVED SURFACES	POINTS OR V'S
ELSE	88
1 SIMPLE AND 1 COMPOUND CURVED SURFACE	POINTS OR V'S
ELSE	89
1 FLAT AND 1 COMPOUND CURVED SURFACE	POINTS OR V'S
ELSE	90
OTHER	POINTS OR V'S
ELSE	91
SHAPE 2 SURFACES	92
ALL CURVED SURFACES SIMPLE CURVES	POINTS OR V'S
ELSE	93
ALL CURVED SURFACES COMPOUND CURVES	POINTS OR V'S
ELSE	94
OTHER	POINTS OR V'S
ELSE	95



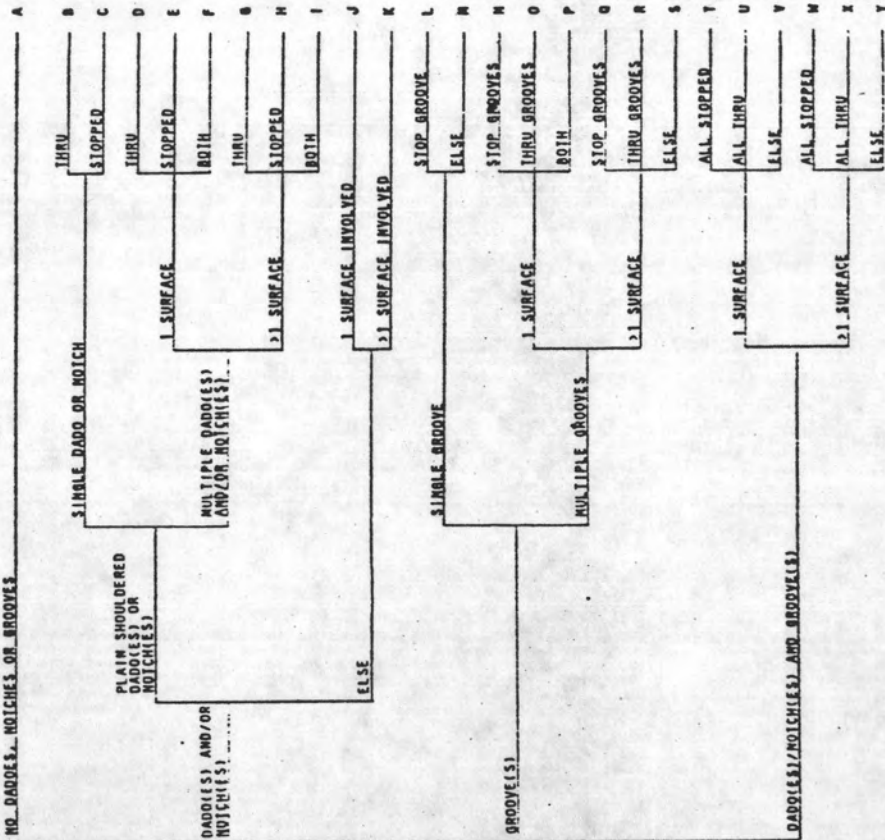
NOTE: A SURFACE IS CONSIDERED SHAPED IF AFTER SHAPING, THE CROSS SECTIONAL GEOMETRY OF THE PART HAS BEEN ALTERED. (E.G. BARDSAW, ALTERNATIVE PATTERNS ARE NOT CONSIDERED SHAPED SURFACES.)

15 16  
CRIMMANNMANN

NOTE: A SHAPED SURFACE MAY BE A SQUARE KNIFE SURFACE THAT REQUIRES SQUARE KNIFING.

NOTE: CARVING IS NOT CONSIDERED TO BE SHAPED SURFACES. IT IS A DECLASSIFIED SURFACE AND IS HANDLED IN TAXONOMY 612.

**CURVED RAIL TAXONOMY #10  
FORM FEATURES -  
DADDIES, NOTCHES AND GROOVES**

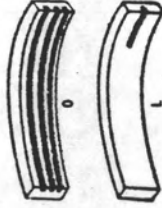
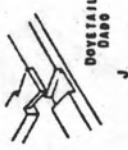
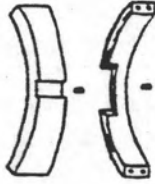


NOTE: A NOTCH IS EQUIVALENT TO A DADDY. NORMALLY, A NOTCH OCCURS ON THE TOP SURFACE OF THE RAIL. HOWEVER, A DADDY OCCURS ON THE FACE OF A RAIL.

NOTE: A DADDY MUST RUN IN A CROSS GRAIN DIRECTION.

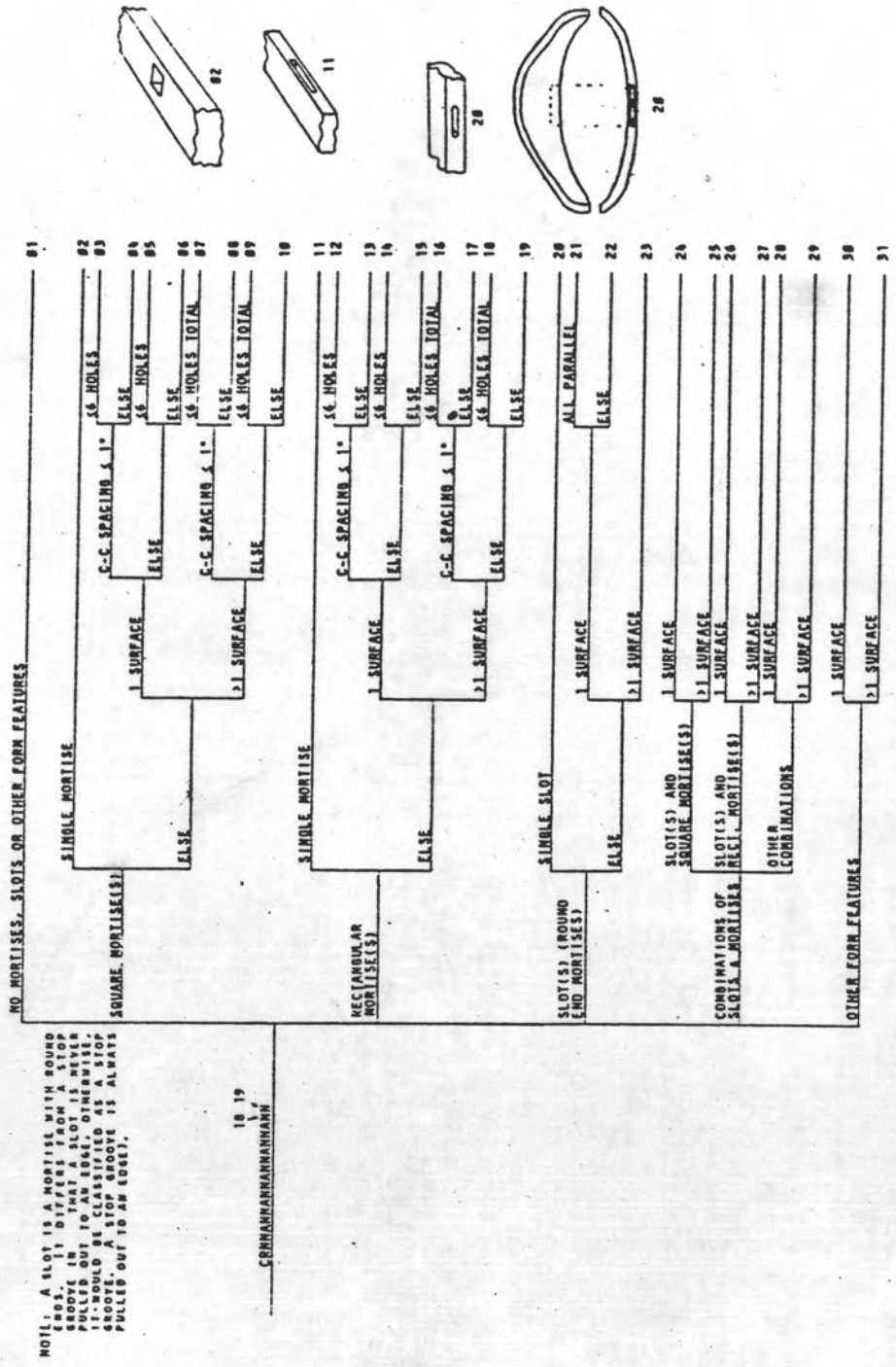
17  
CRRRHHHHHHHHHH

NOTE: A STOP GROOVE MUST BE PULLED OUT TO AN EDGE. OTHERWISE, IT IS A SLOT.



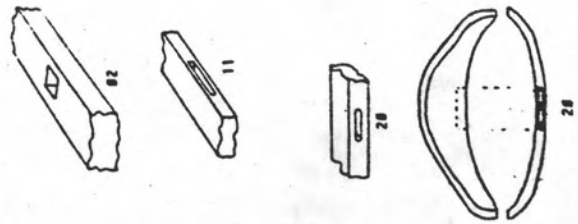


CURVED RAIL TAXONOMY #11  
MORTISES, SLOTS AND OTHER  
FORM FEATURES

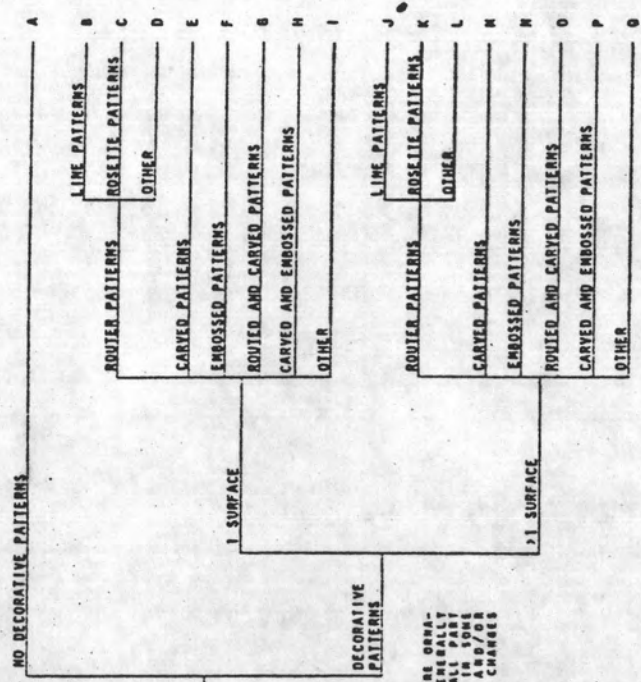


NOTE: A SLOT IS A MORTISE WITH ROUND ENDS. IT DIFFERS FROM A STOP GROOVE IN THAT A SLOT IS NEVER PULLED OUT TO AN EDGE. OTHERWISE, IT WOULD BE CLASSIFIED AS A STOP GROOVE. A STOP GROOVE IS ALWAYS PULLED OUT TO AN EDGE.

18 19  
1 1  
CERNIKHANNMANNHANN

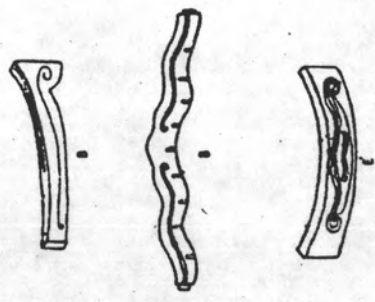


CURVED RAIL TAXONOMY #12  
DECORATIVE PATTERNS

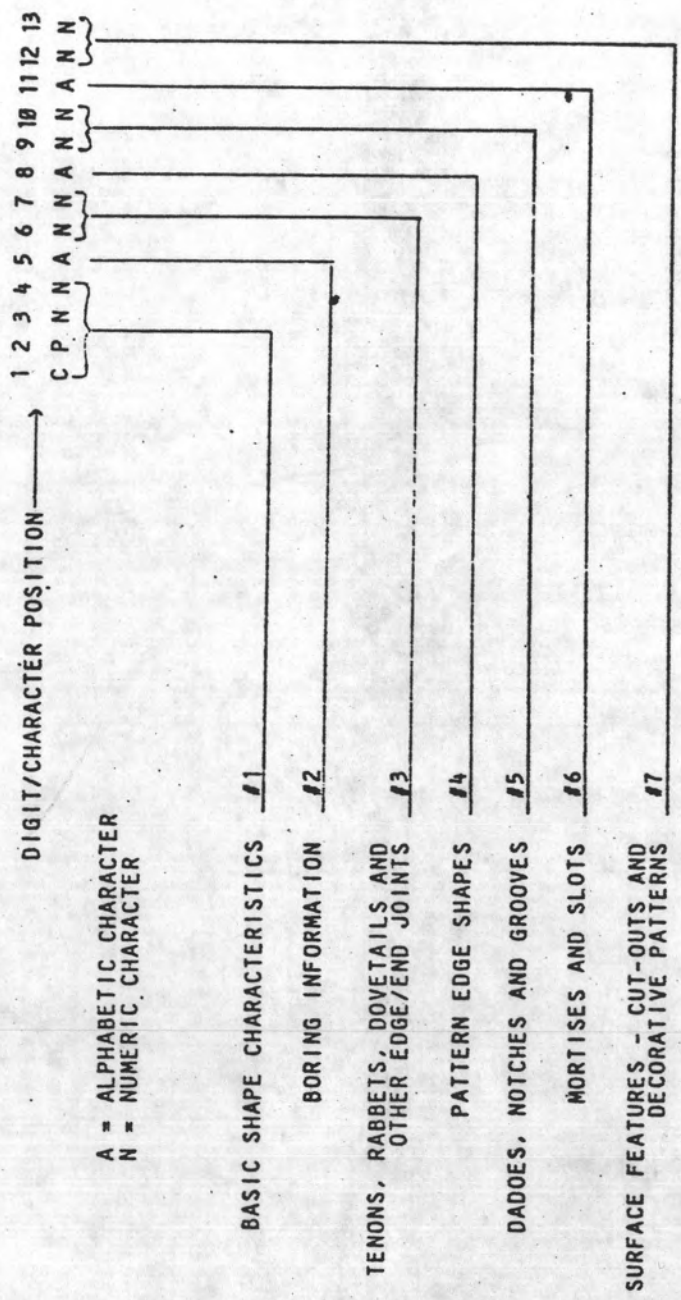


28  
SERMANHANMANHANNA

NOTE: DECORATIVE PATTERNS ARE ORNAMENTAL FEATURES THAT GENERALLY DO NOT CHANGE THE OVERALL PART CLASSIFICATION. CARVING, AND/OR EMBOSSING, DIMENSIONAL CHANGES MAY OCCUR.

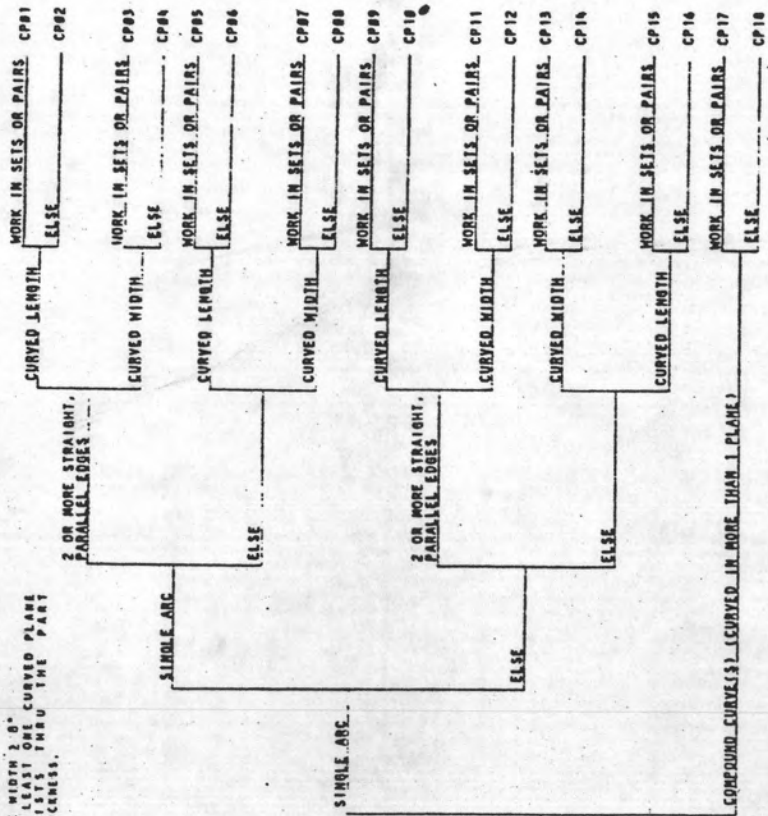
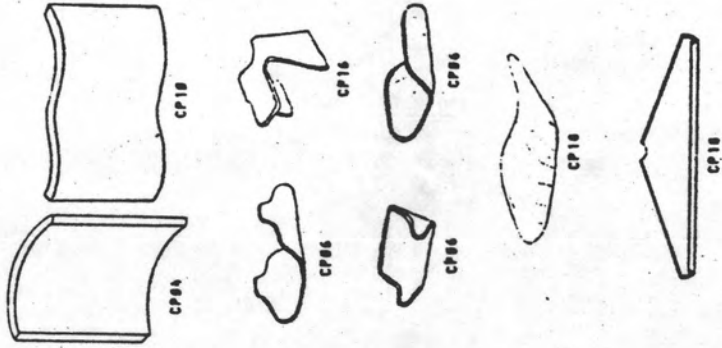


CURVED PANEL TAXONOMIES



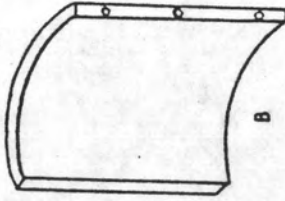
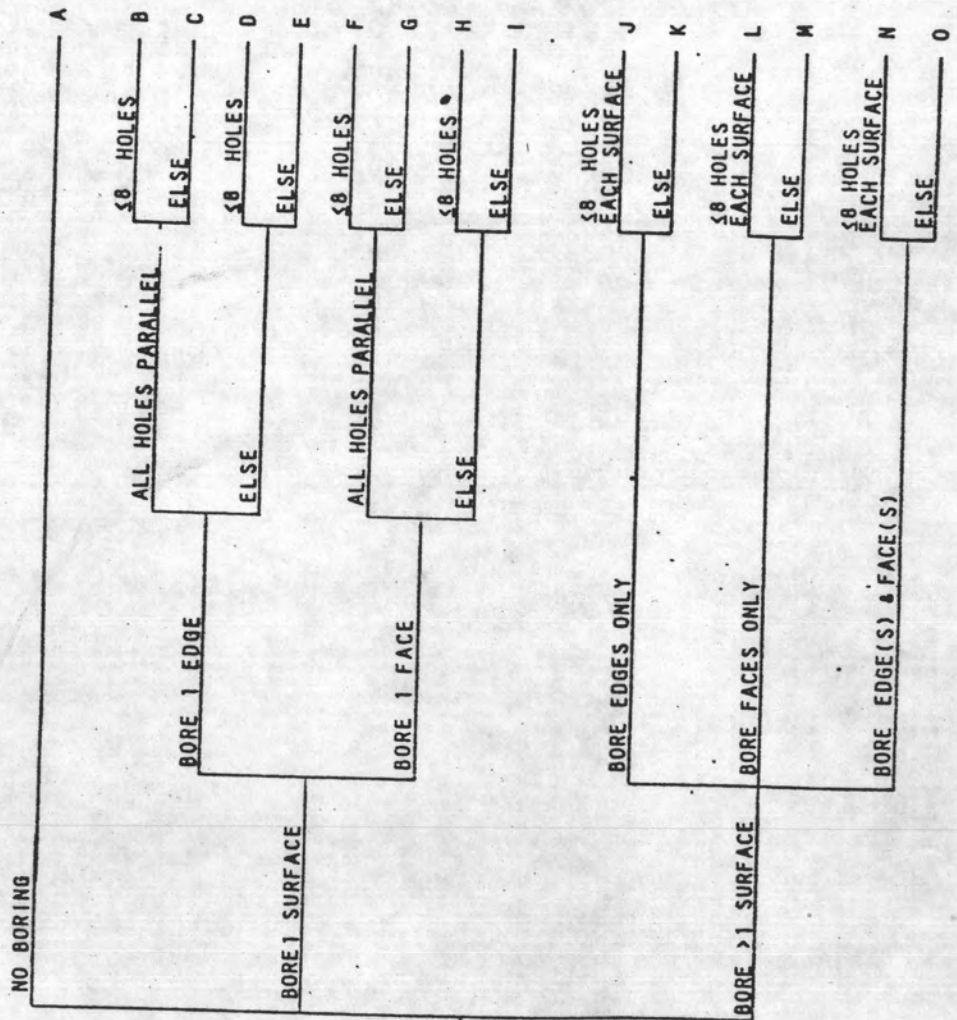
CURVED PANEL TAXONOMY #1  
BASIC SHAPE CHARACTERISTICS

NOTE: A CURVED PANEL IS A PART SUCH THAT:  
1. THE WIDTH IS 0°  
2. AT LEAST ONE CURVED PLANE EXISTS THRU THE PART THICKNESS.



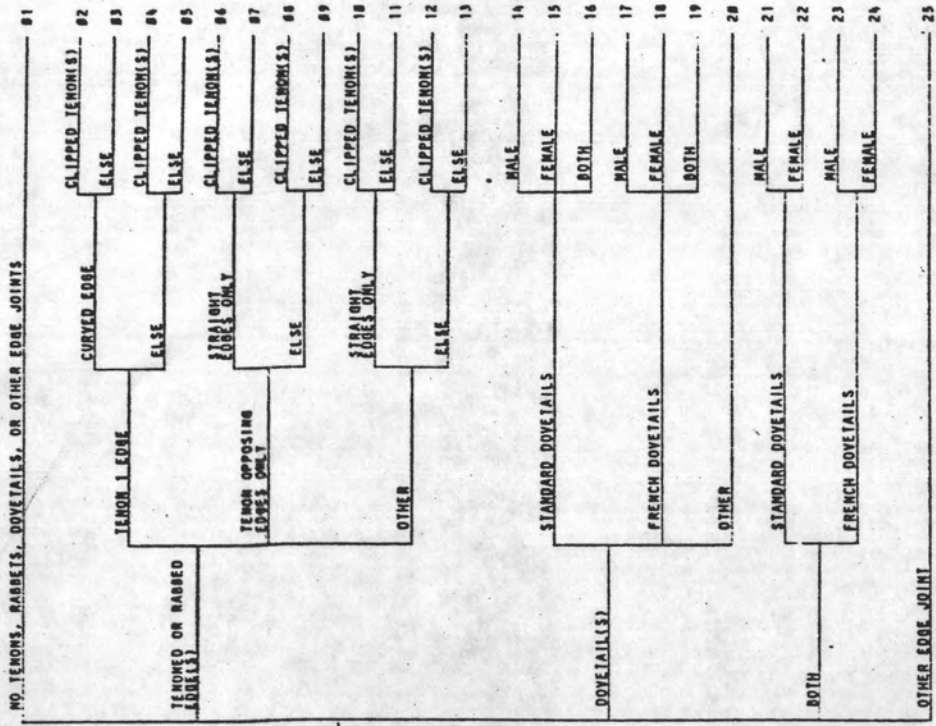
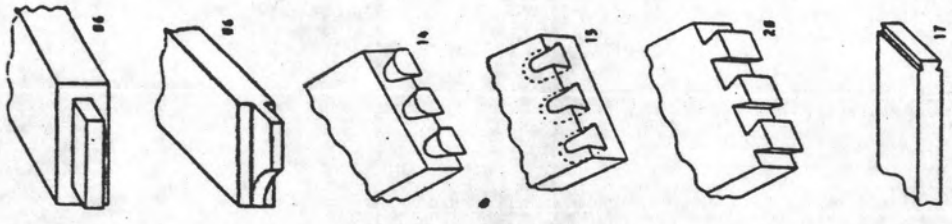
--- CURVED PANELS ---

CURVED PANEL TAXONOMY #2  
BORING INFORMATION



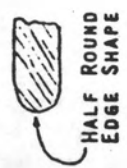
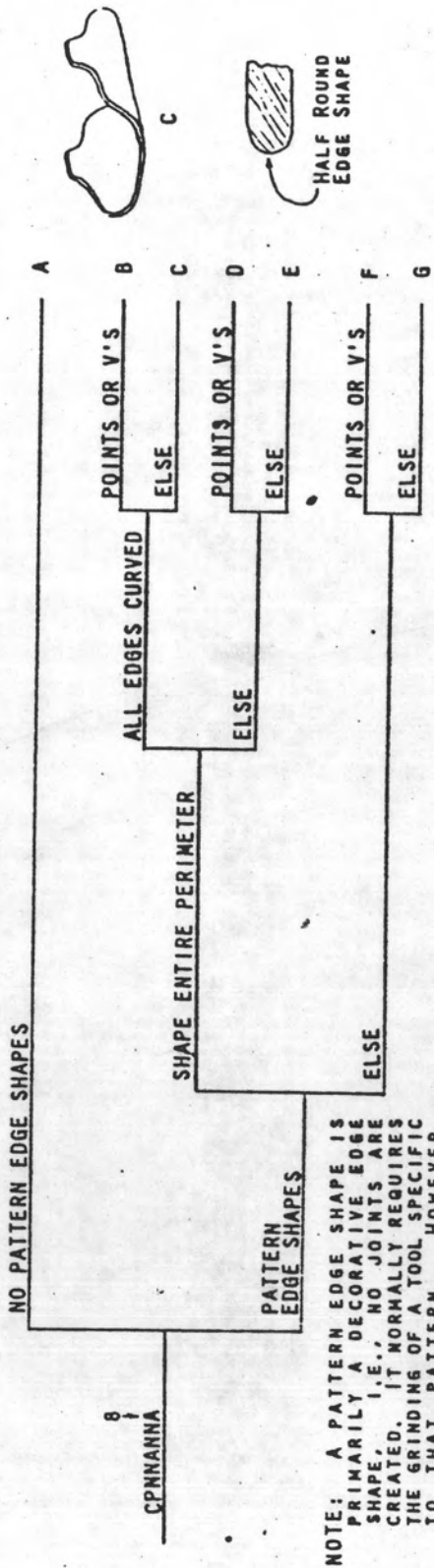
5  
CPNNA

CURVED PANEL TAXONOMY #3  
EDGE JOINTS  
TENONS, RABBETS, DOVETAILS  
OR OTHER EDGE-END JOINTS



67  
CPHMAN

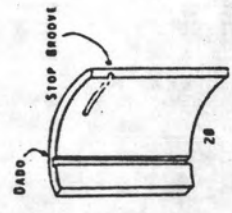
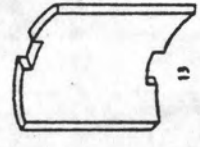
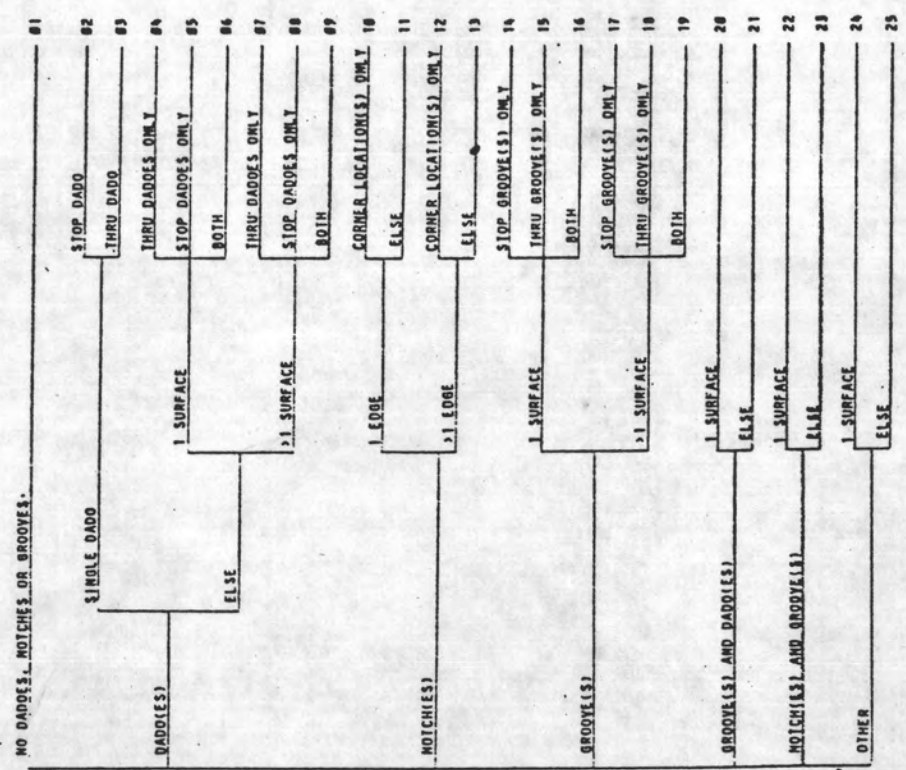
CURVED PANEL TAXONOMY #4  
PATTERN EDGE SHAPES



NOTE: A PATTERN EDGE SHAPE IS PRIMARILY A DECORATIVE EDGE SHAPE. I.E., NO JOINTS ARE CREATED. IT NORMALLY REQUIRES THE GRINDING OF A TOOL SPECIFIC TO THAT PATTERN. HOWEVER, SQUARE-KNIFED EDGES ARE CONSIDERED PATTERN SHAPED EDGES.

8  
CPNANNA

CURVED PANEL TAXONOMY #5  
DADDES, NOTCHES AND GROOVES



9 10  
CPMANNANN

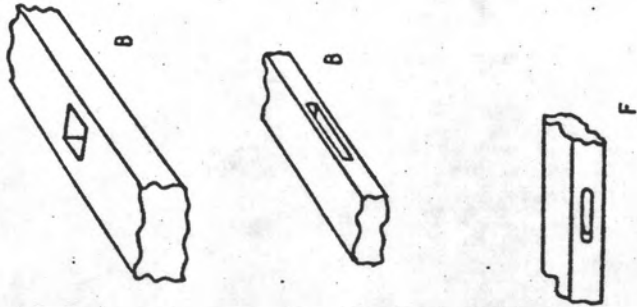
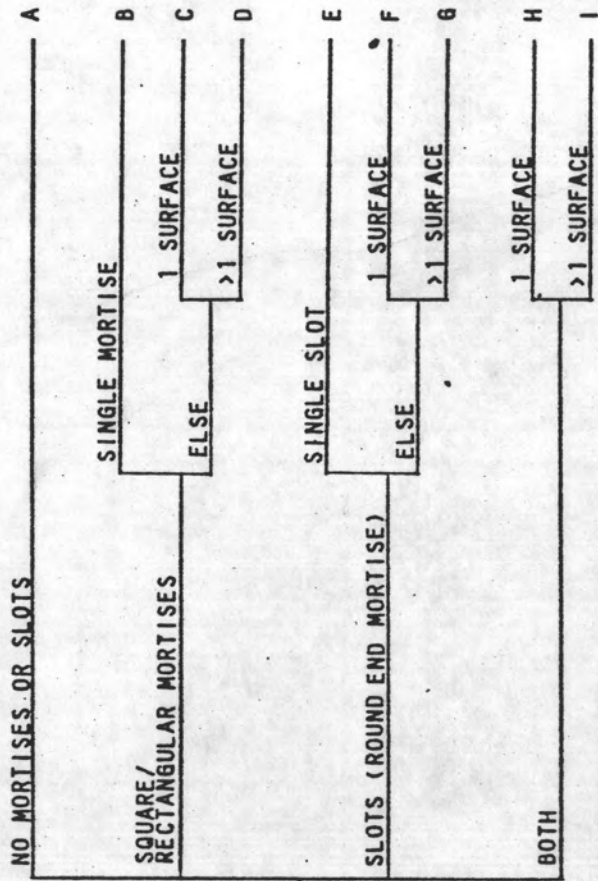
NOTE: A DADO MUST RUN IN A CROSS GRAIN DIRECTION EITHER PERPENDICULAR TO, OR DIAGONALLY ACROSS, THE GRAIN.

NOTE: A NOTCH NORMALLY OCCURS ON THE EDGE OF A PANEL AND IS CUT THRU THE THICKNESS OF THE PART.

NOTE: A STOP GROOVE RUNS PARALLEL TO THE GRAIN AND IS PULLED OUT AN EDGE.



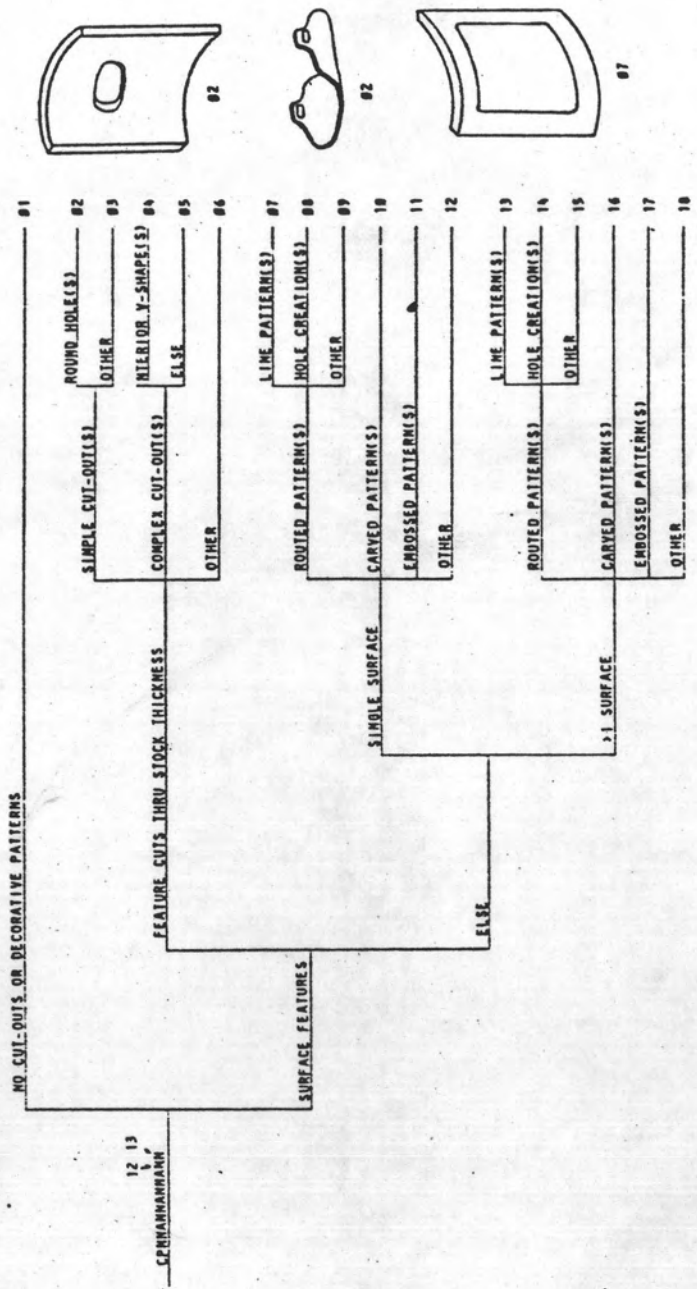
**CURVED PANEL TAXONOMY #6**  
**MORTISES AND SLOTS**



11  
CPNANNANNA

NOTE: A SLOT IS NEVER PULLED OUT TO AN EDGE, OTHERWISE IT WOULD BE CLASSIFIED AS A STOP GROOVE. A SLOT MAY RUN PARALLEL OR PERPENDICULAR TO THE GRAIN DIRECTION. IT IS DISTINGUISHED FROM A DADO BY ITS ROUND ENDS.

CURVED PANEL TAXONOMY #7  
SURFACE FEATURES  
CUT-OUTS AND DECORATIVE PATTERNS



12 13  
SPRIANMANN

IRREGULAR PARTS TAXONOMY

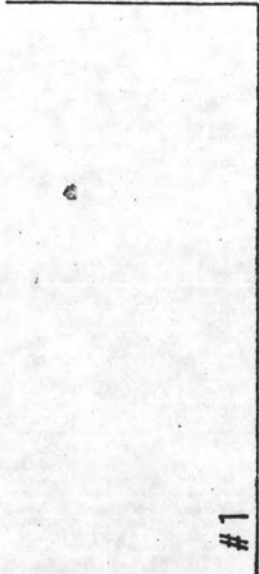
1 2 3 4  
| R N N

DIGIT/CHARACTER POSITION

A = ALPHABETIC CHARACTER  
N = NUMERIC CHARACTER

IRREGULAR PARTS

#1



IRREGULAR PARTS TAXONOMY

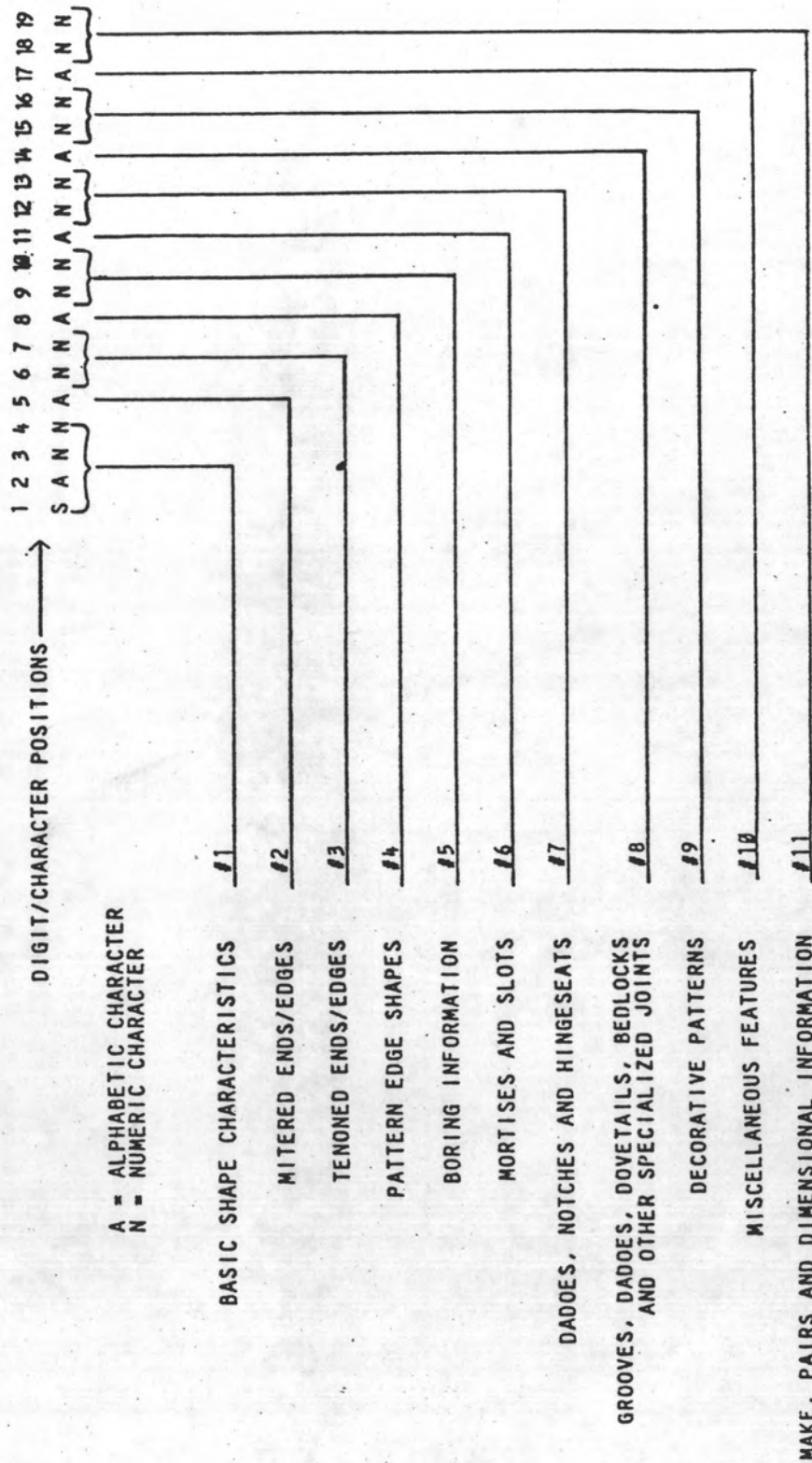
1 2 3 4  
| R N N

DIGIT/CHARACTER POSITION

A = ALPHABETIC CHARACTER  
N = NUMERIC CHARACTER

IRREGULAR PARTS #1

SUB-ASSEMBLY TAXONOMIES



SUB-ASSEMBLY TAXONOMY #1  
BASIC SUB-ASSEMBLY TYPES

NOTE: A SUB-ASSEMBLY IS A COMBINATION OF COMPONENT PARTS THAT REQUIRES MACHINING OPERATIONS SUBSEQUENT TO BECOMING A SUB-ASSEMBLY.

NOTE: THE SUB-ASSEMBLY TAXONOMIES ARE USED TO ENCODE ALL PART GEOMETRIES AND FORM FEATURES THAT CAN BE COMBINED INTO A SUB-ASSEMBLY.

OPEN FRAMES

EXAMPLES:  
FRAME DOORS & ENDS  
MIRROR FRAMES  
GLASS TOP TABLE TOPS  
CHAIR BACK S/A'S

RAILS/PANELS WITH ATTACHED MOULDING AND/OR OVERLAYS

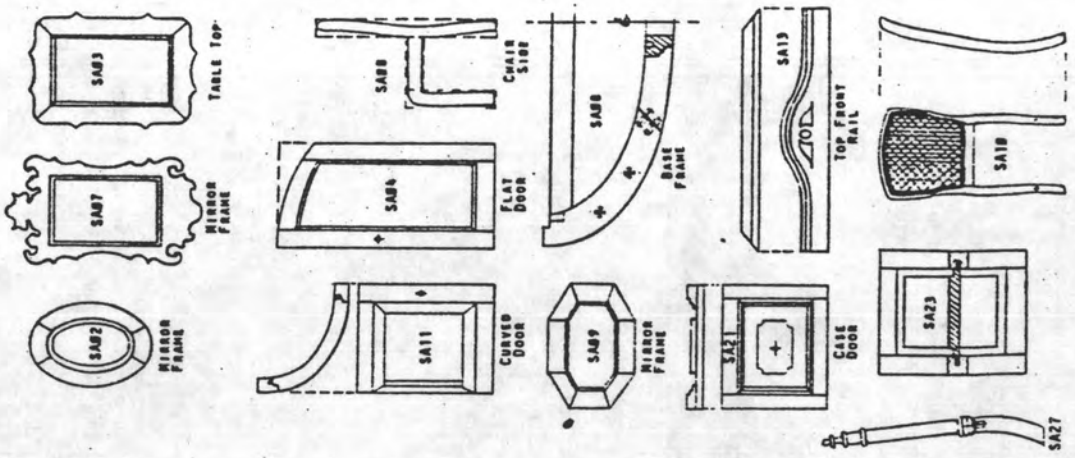
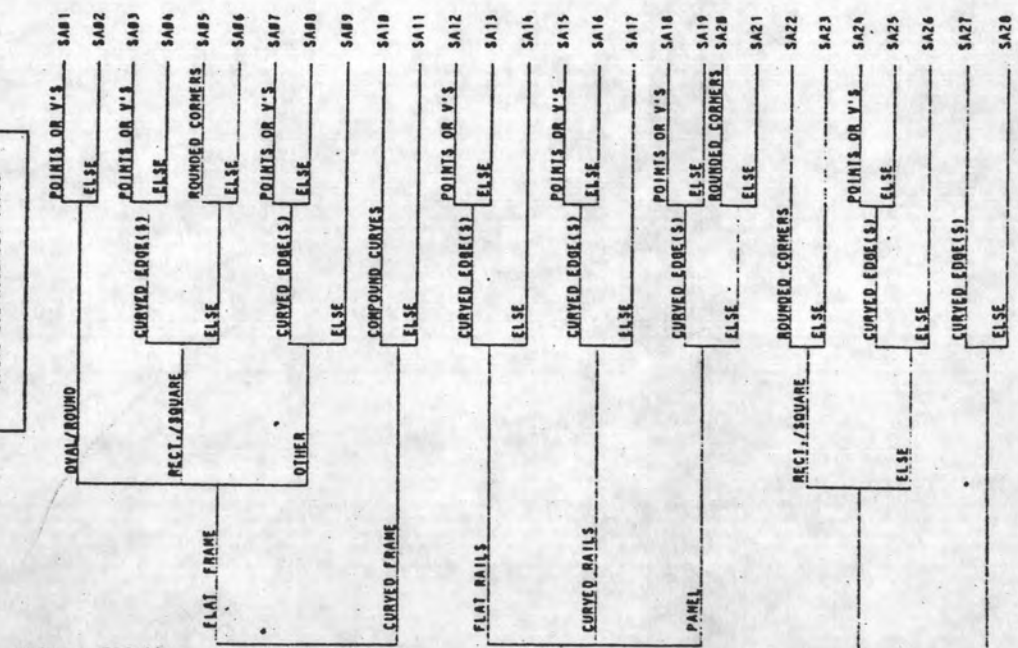
EXAMPLES:  
DOORS & DRAWER FRONTS  
MIRROR FRAMES  
MOULDED TOP FRONT RAILS  
PANEL ON FRAME ENDS  
PANEL ON FRAME TOPS

SUB-ASSEMBLIES

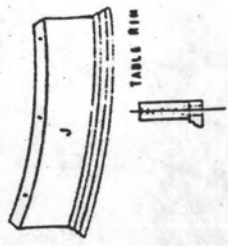
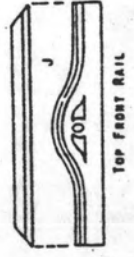
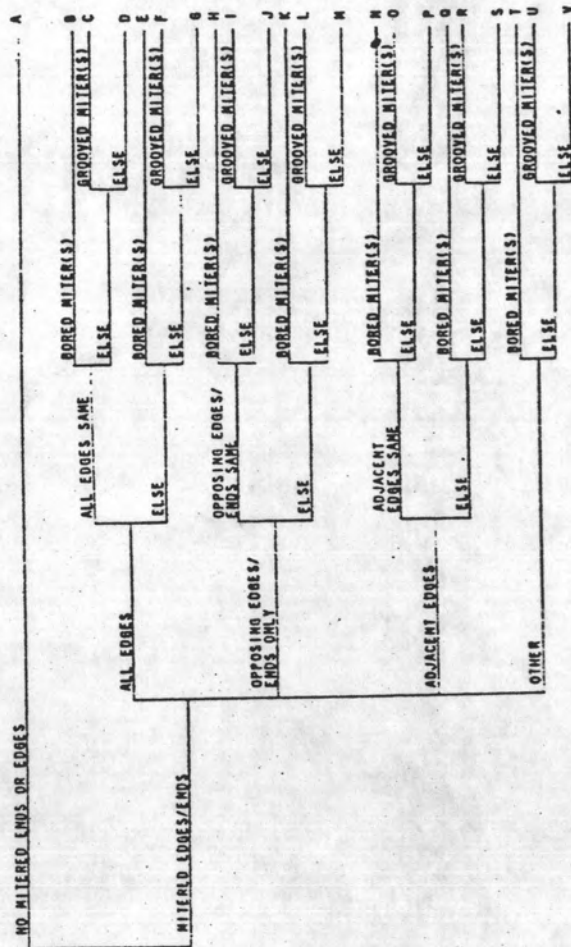
FILLED FRAMES (CENTER PANEL FRAMES)

EXAMPLES:  
RAISED PANEL DOORS  
RAISED PANEL END PANELS

OTHER

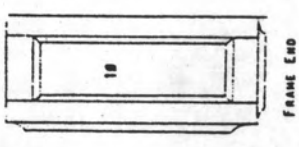
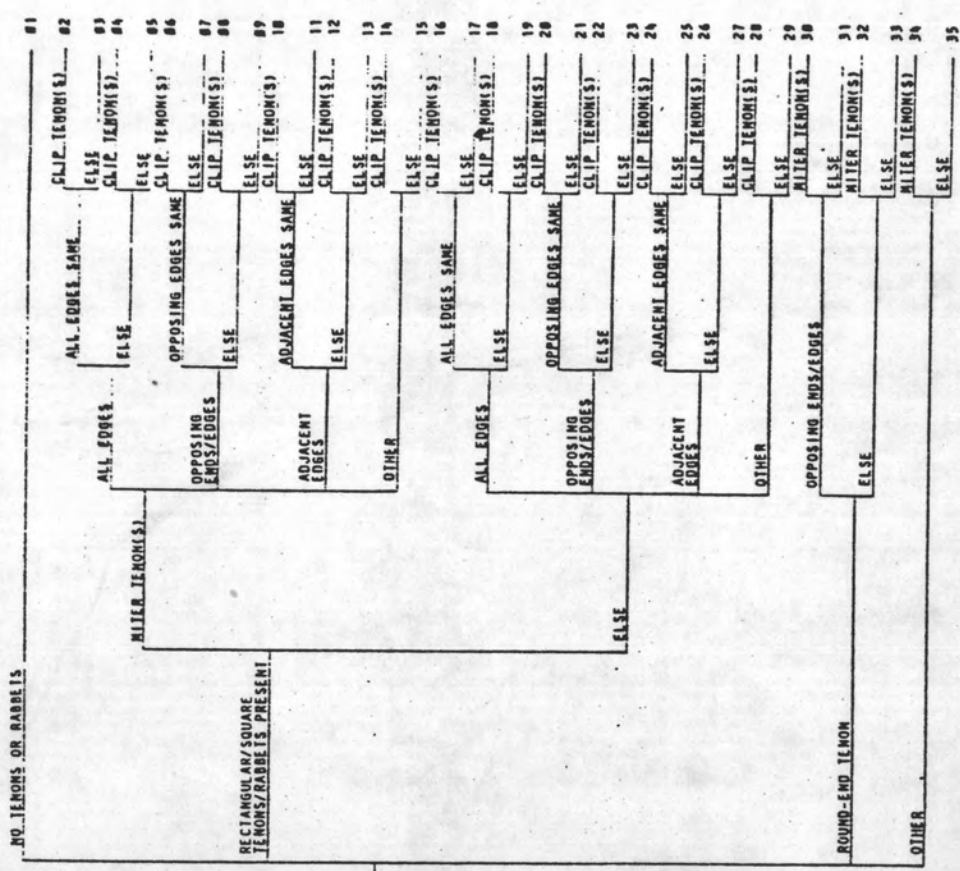


SUB-ASSEMBLY TAXONOMY #2  
MITERED ENDS AND/OR EDGES



SANWA

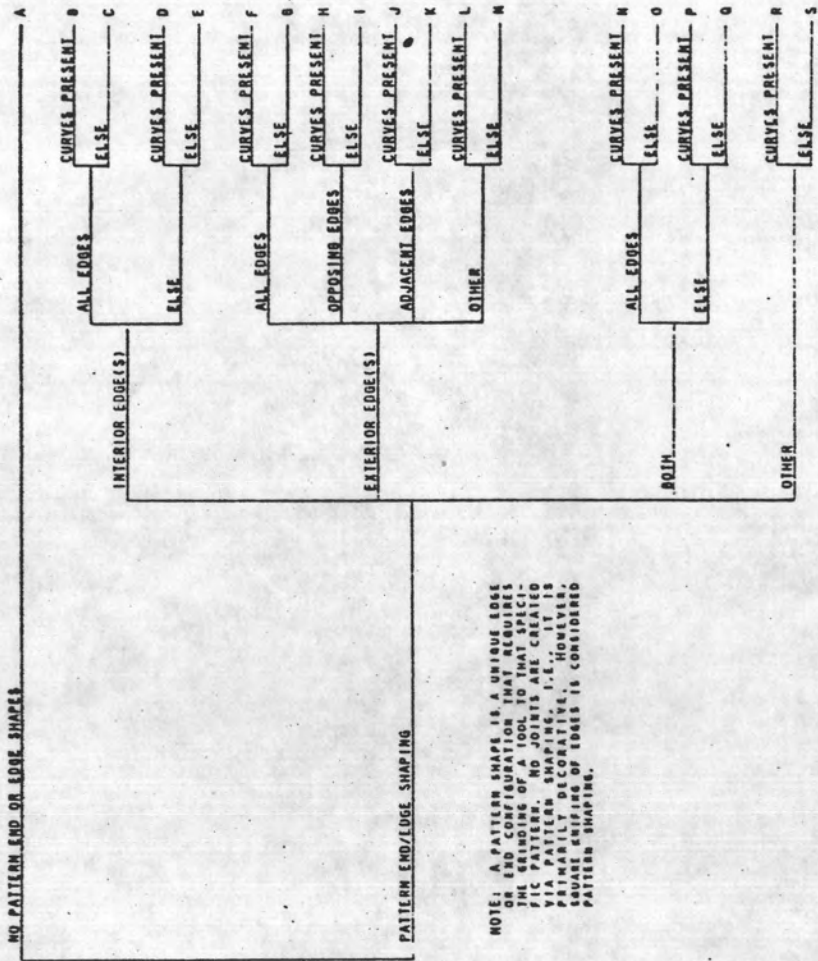
SUB-ASSEMBLY TAXONOMY #3  
TENONED ENDS/EDGES



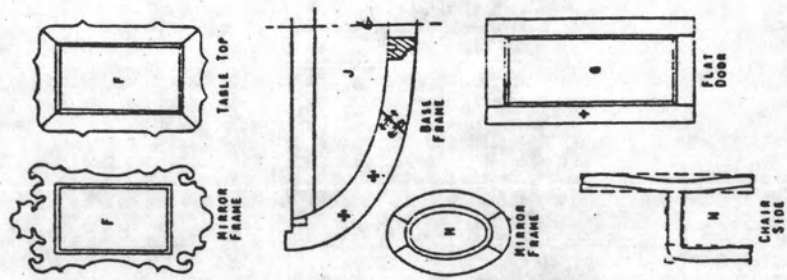
67  
SANNMANN



SUB-ASSEMBLY TAXONOMY #A  
PATTERN END/EDGE SHAPES

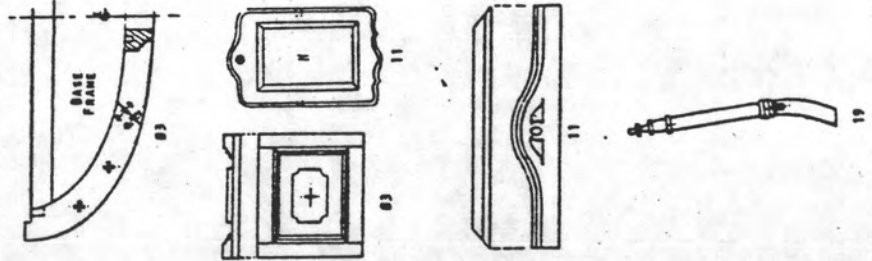
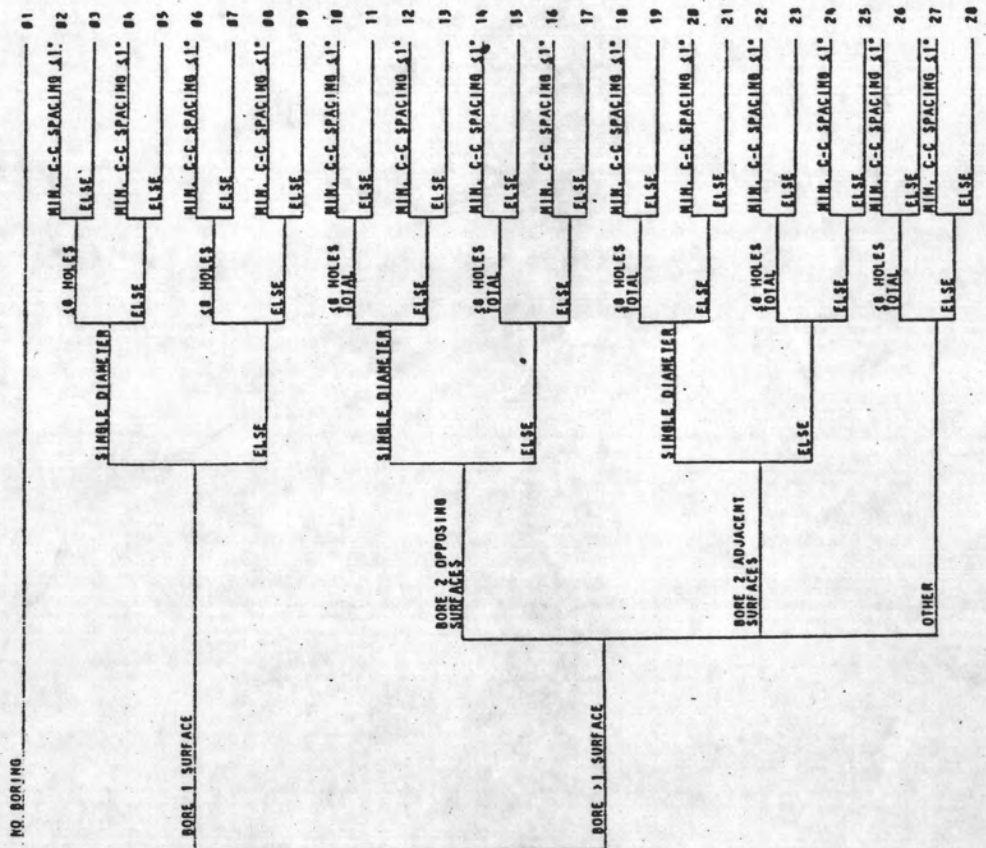


NOTE: A PATTERN SHAPE IS A UNIQUE EDGE OR END CONFIGURATION THAT REQUIRES THE GRINDING OF A TOOL TO THAT SPECIFIC PATTERN. NO JOINTS ARE CREATED BETWEEN PATTERN SHAPES, I.E. HOWEVER, SQUARE GRINDING OF TOOLS IS CONSIDERED PATTERN SHAPING.



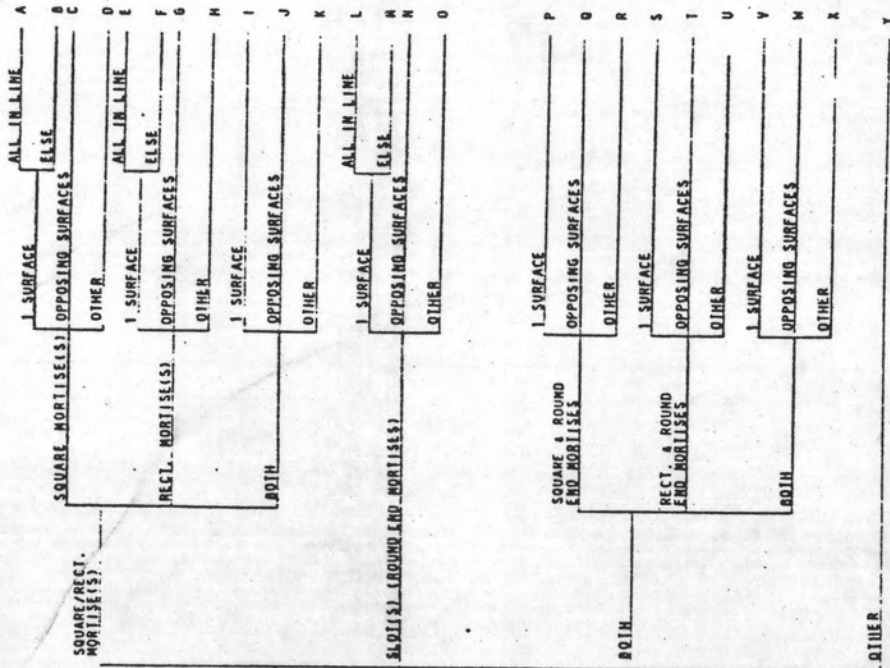
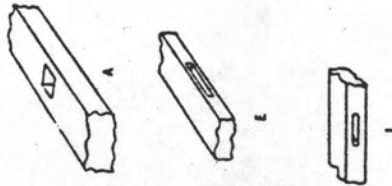
SARNAWA

SUB-ASSEMBLY TAXONOMY #5  
BORING INFORMATION



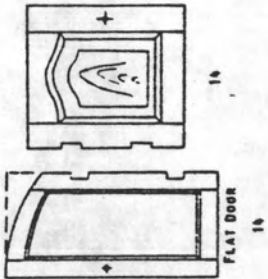
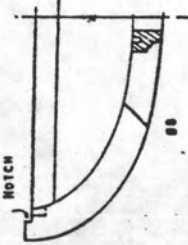
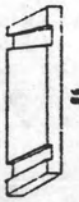
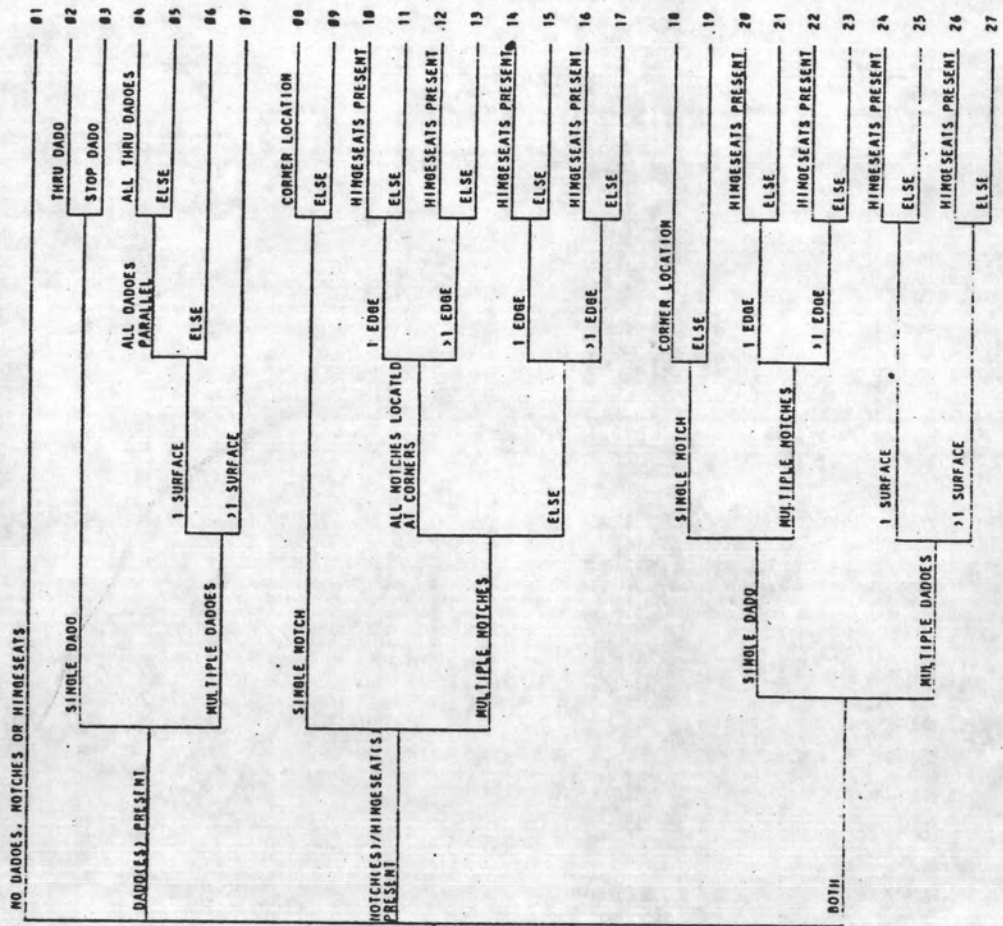
9 19  
SABHANNANN

SUB-ASSEMBLY #6  
FORM FEATURES  
MORTISES AND SLOTS



11  
SANNANNA

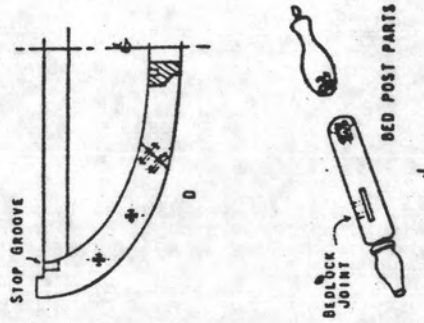
SUB-ASSEMBLY TAXONOMY #7  
DADOES, NOTCHES AND HINGESEATS



12 13  
SANWANNHANN

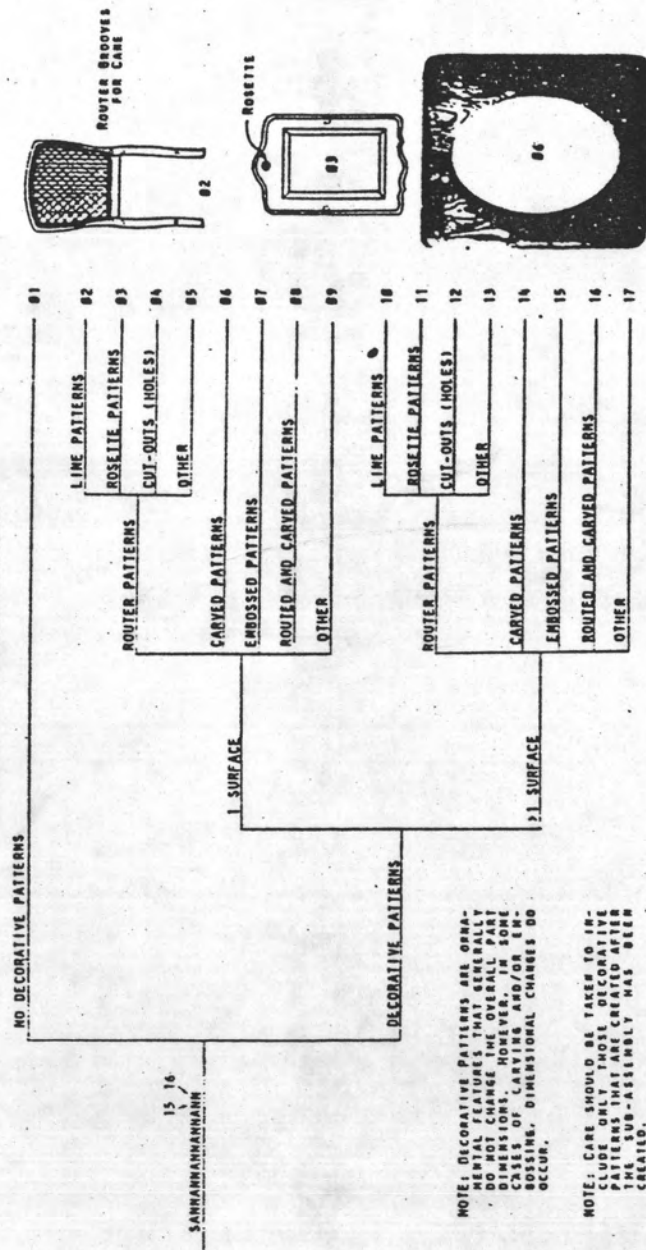
SUB-ASSEMBLY #8  
GROOVES, DOVETAILS, BEDLOCKS  
AND OTHER SPECIALIZED JOINTS

NO. GROOVES, DOVETAILS, BEDLOCKS OR OTHER SPECIALIZED JOINTS PLAIN GROOVE(S) DOVETAILS(S) BEDLOCK JOINT(S) PLAIN GROOVE(S) AND DOVETAIL(S) PLAIN GROOVE(S) AND BEDLOCK JOINT(S) OTHER JOINTS	A B C D E F G H I J K L M N O P Q
--	---



14  
SAMMANMANNA

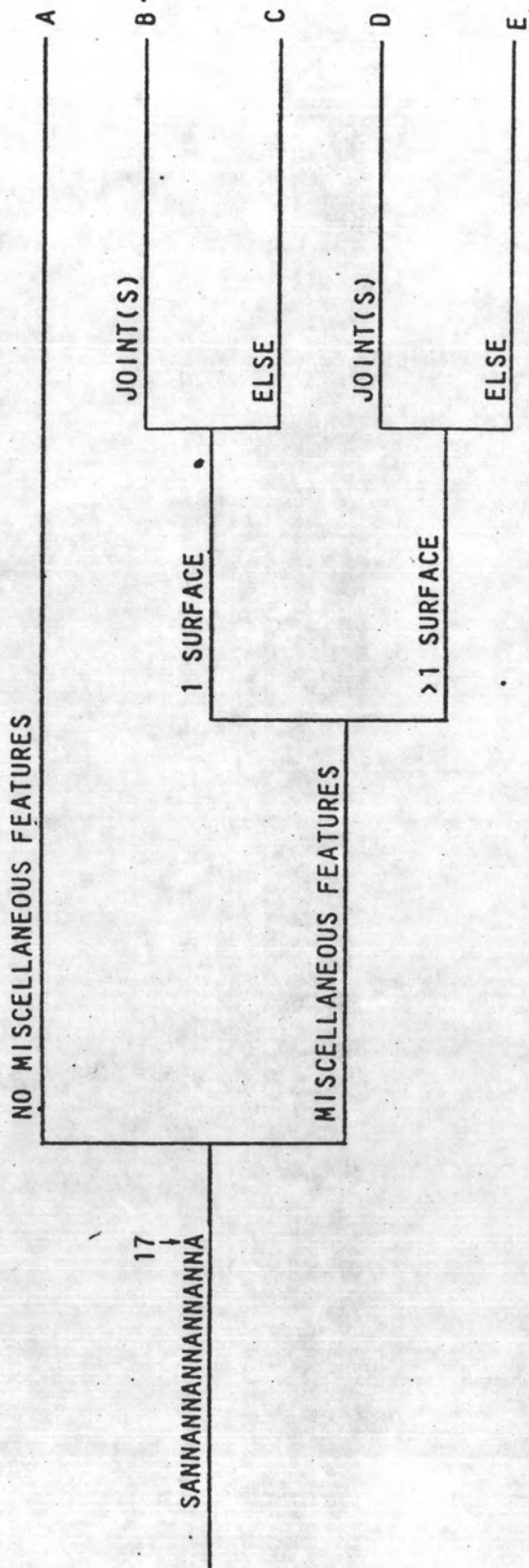
SUB-ASSEMBLY TAXONOMY #9  
DECORATIVE PATTERNS



NOTE: DECORATIVE PATTERNS ARE ORNAMENTAL FEATURES THAT GENERALLY DO NOT CHANGE THE OVERALL PART DIMENSIONS. HOWEVER, IN SOME CASES OF LAYERS, HOLES, BOSSINGS, DIMENSIONAL CHANGES DO OCCUR.

NOTE: CARE SHOULD BE TAKEN TO INCLUDE ONLY THOSE DECORATIVE PATTERNS THAT ARE CREATED AFTER SUB-ASSEMBLY HAS BEEN CREATED.

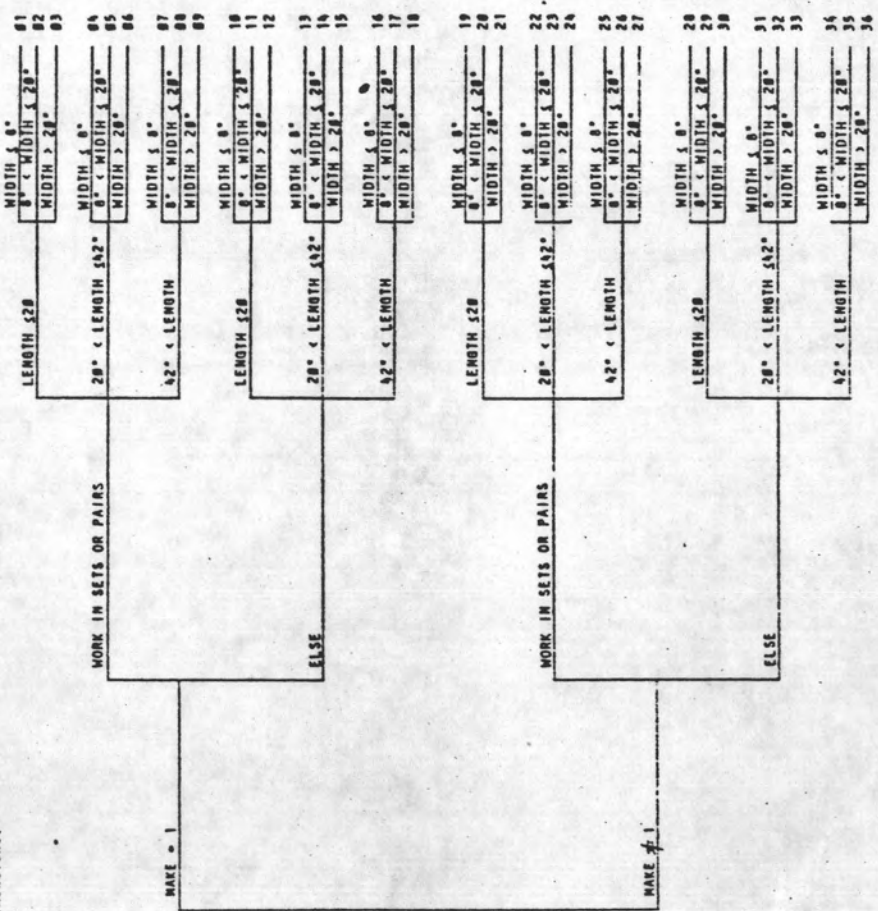
SUB-ASSEMBLY TAXONOMY #10  
MISCELLANEOUS FEATURES



17  
SANNANNANNANNA

SUB-ASSEMBLY TAXONOMY #11  
MAKE PAIRS AND  
DIMENSIONAL INFORMATION

NOTE: DIMENSIONAL INFORMATION IN THIS TAXONOMY PERTAINS TO OVERALL SUB-ASSEMBLY DIMENSIONS.



10 19  
SAMMANNHANNHANN



ภาคผนวก ค.

ไดคัทใช้แทนเครื่องจักรในการผลิตเฟอร์นิเจอร์

โค้ดที่ใช้แทนเครื่องจักรในการผลิตเฟอร์นิเจอร์ (OPCODES)

1. แผนกเลื่อย

CS01	ตัดหยาบ (Cut off Saw)
CS02	เลื่อยผ่า (Rip Saw)
CS03	เลื่อยคว้าน (Band Saw)
CS04	ตัดมุม 45 (Angle Saw)
CS05	ตัดละเอียด (Radial Saw)
CS06	เลื่อยผ่าซุง (Gang Rip Saw)
CS07	เลื่อยฉลุ (Jig Saw)

2. แผนกไส

PM01	ไสสี่หน้า (Four Side Planer)
PM02	ไสสองหน้า (Double Side Planer)
PM03	ไสขีด (Jointer or Surface Planer)

3. แผนกเจาะ/ปอกเดือย

JM01	ทำเดือยและตัดเดือยกลม (Dowel & Dowel Chamfer Machine)
JM02	ปอกเดือยรูปไข่ (Tenoner)
JM03	ปอกเดือยรูปไข่ 2 หัว (Double Tenoner)
JM04	เจาะร่องรูปไข่ (Jointing Machine)
JM05	เจาะร่องรูปไข่หลายหัว (Multi-Jointing Machine)
JM06	เจาะแนวตั้งหลายหัว (Vertical Multi-Drill)
JM07	เจาะแนวนอน (Horizontal Drill)

4. แผนกขึ้นรูป

- ML01 ตีบัววงใน (Router)
- ML02 ตีบัววงนอก (Moulder)
- ML03 เครื่องกลึง (Turning lathe)
- ML04 กลึงลอกแบบ (Copy lathe)
- ML05 เครื่องกลึงลอกแบบซอปปเปอร์ (Copy Carver)
- ML06 เครื่องกลึงสูญญากาศ (Copy lathe Vacuum)
- ML07 เครื่องเหลากกลม (Round Pole Milling)

5. แผนกขัดผิว

- SM01 ขัดสายพานตั้ง (Verticle Belt Sander)
- SM02 ขัดสายพานนอน (Horicental Belt Sander)
- SM03 ขัดหลายอย่าง (Universal Belt Sander)
- SM04 ขัดเหลากกลม (Round Pole Sander)
- SM05 ขัดฟองน้ำ (บัวน้ำ) (Verticle Sponge Sander)
- SM06 ขัดโค้ง (Form Sander)
- SM07 ขัดแบบจาน (Disk Sander)
- SM08 ขัดแบบล้อ (Drum Sander)

6. แผนกอัดประกอบ (Press and Assembling Machine)

- PA01 อัดลมแบบตั้งโต๊ะ (Press and Assembling Machine)
- PA02 อัดหมุน (Rolling Press)
- PA03 อัดประกอบเก้าอี้ (Assembling Machine for Chair)
- PA04 ชันเกลียวอัตโนมัติ (Auto nailer)

7. แผนกตกแต่งชิ้นสุดท้าย (Finishing)

- FF01 พ่นสีแบบไฟฟ้า (Electric Spray)
- FF02 พ่นสีแบบใช้ลม (Air Spray)
- FF03 ขนย้ายวัสดุ (Material Handling)
- FF04 ตู้พ่นสี (Spray Booths)
- FF05 อบสี (Drying)

8. ตรวจสอบและซ่อมแซม (Inspection and Repair)

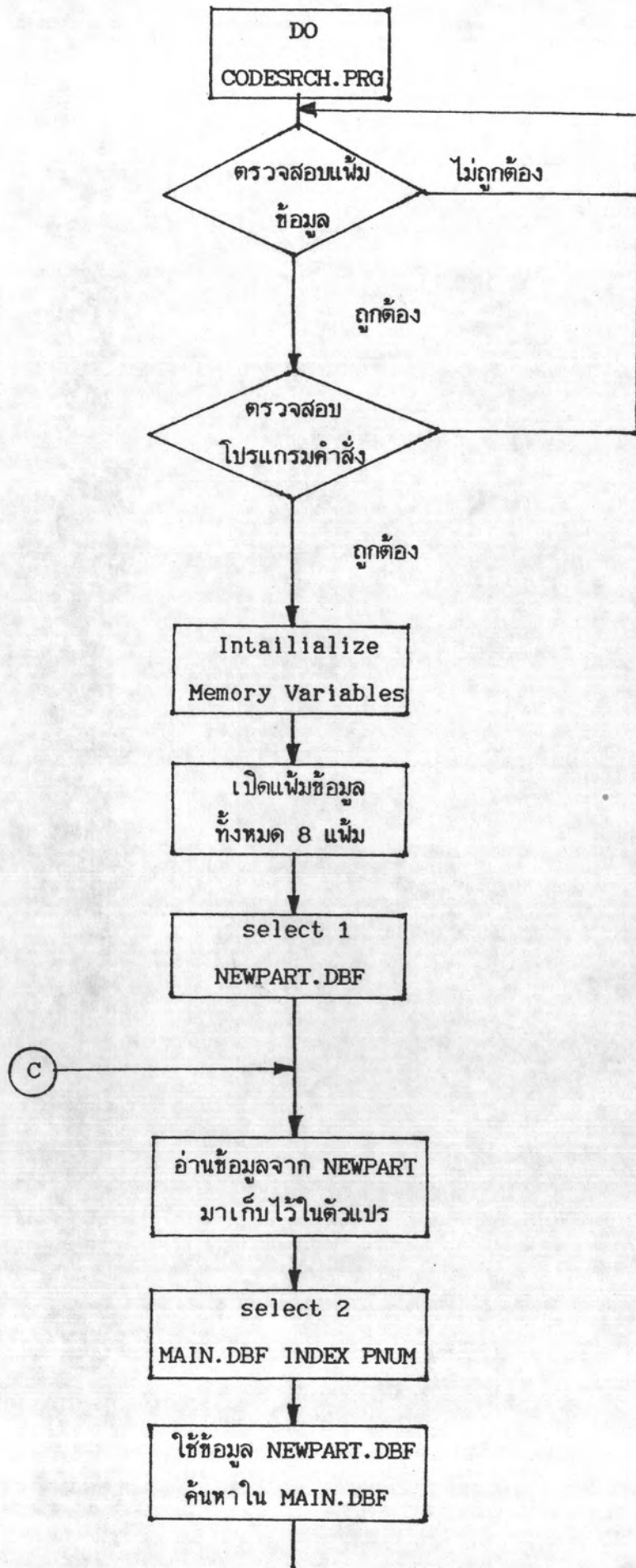
- IR01 ตรวจสอบและซ่อมแซม

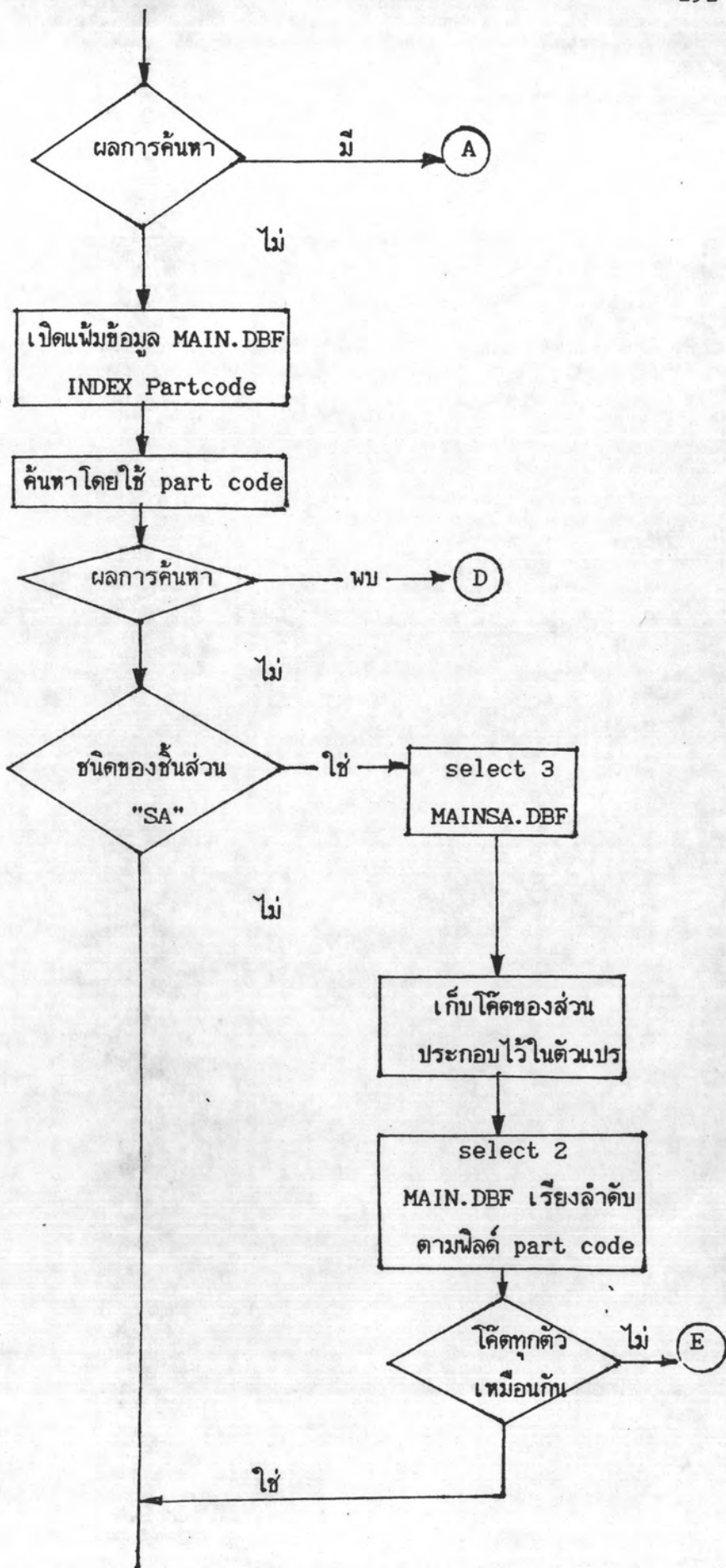
9. ชิ้นส่วนที่ซื้อจากภายนอก (Purchasing Outside)

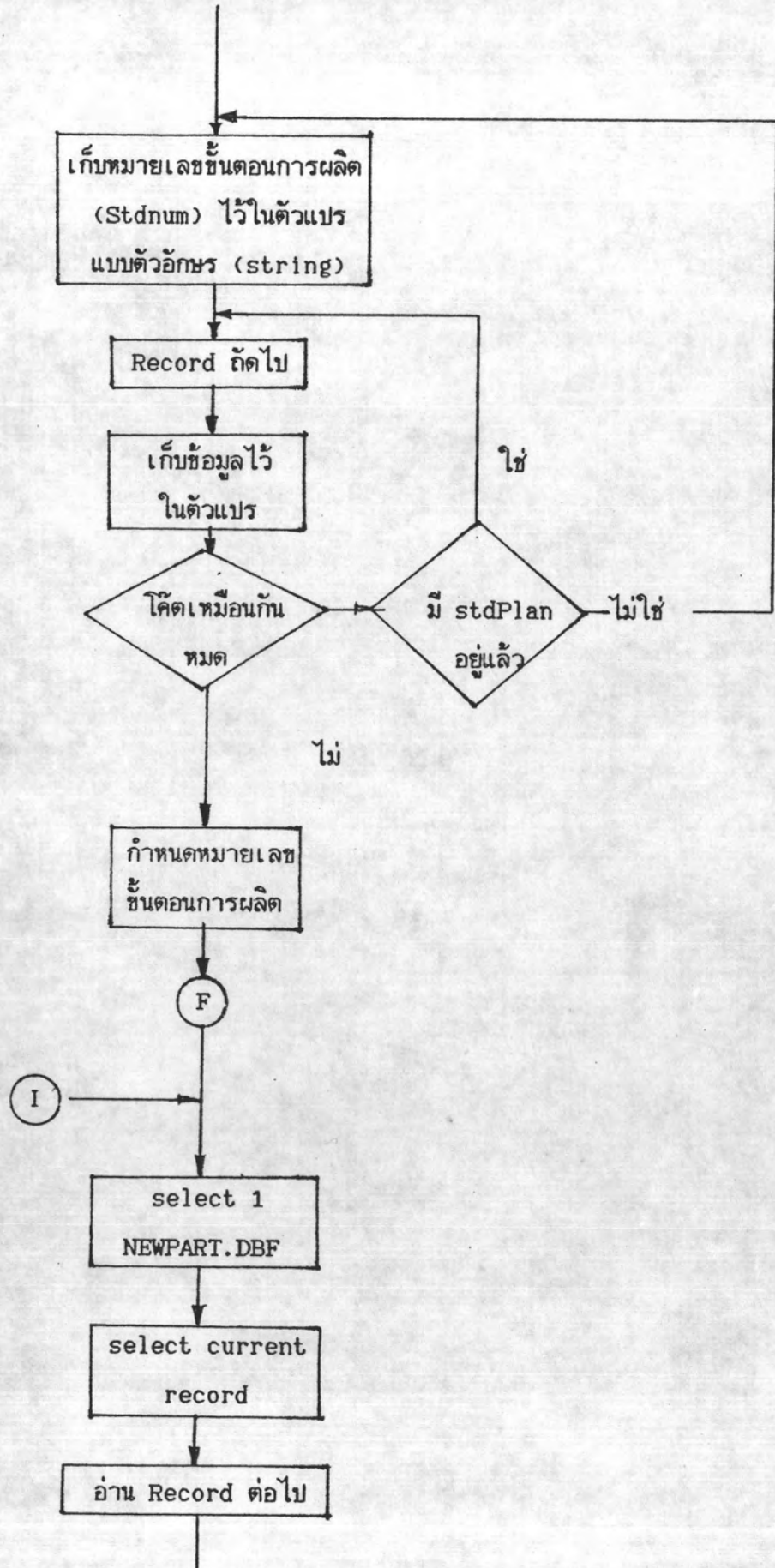
- PC01 ชิ้นส่วนซื้อสำเร็จ (Completely Purchased Component)
- PC02 ชิ้นส่วนซื้อแล้วต้องตกแต่ง (Purchased Component Requiring Some Machining)

ภาคผนวก ง.

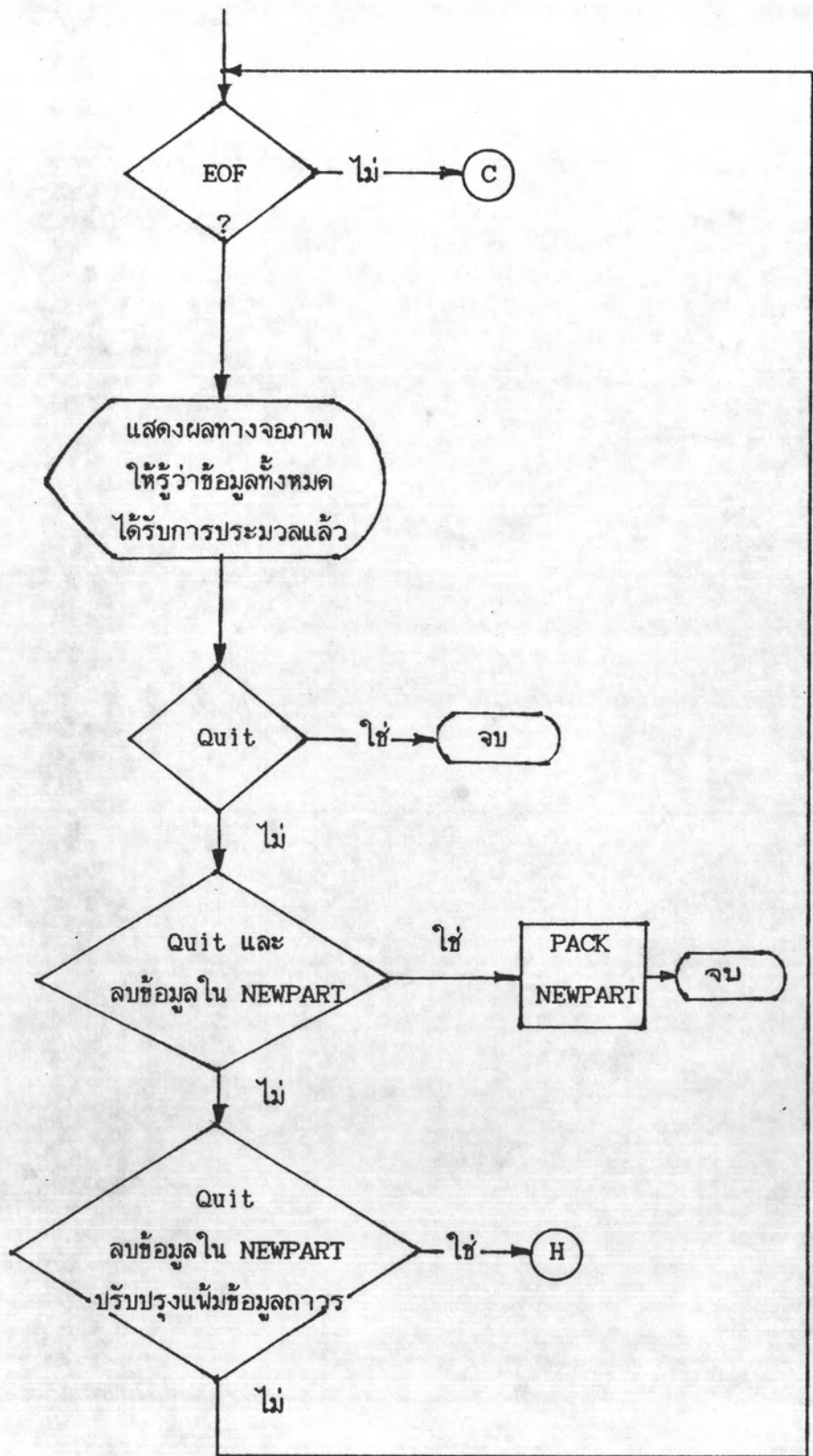
โครงสร้างของ โปรแกรม

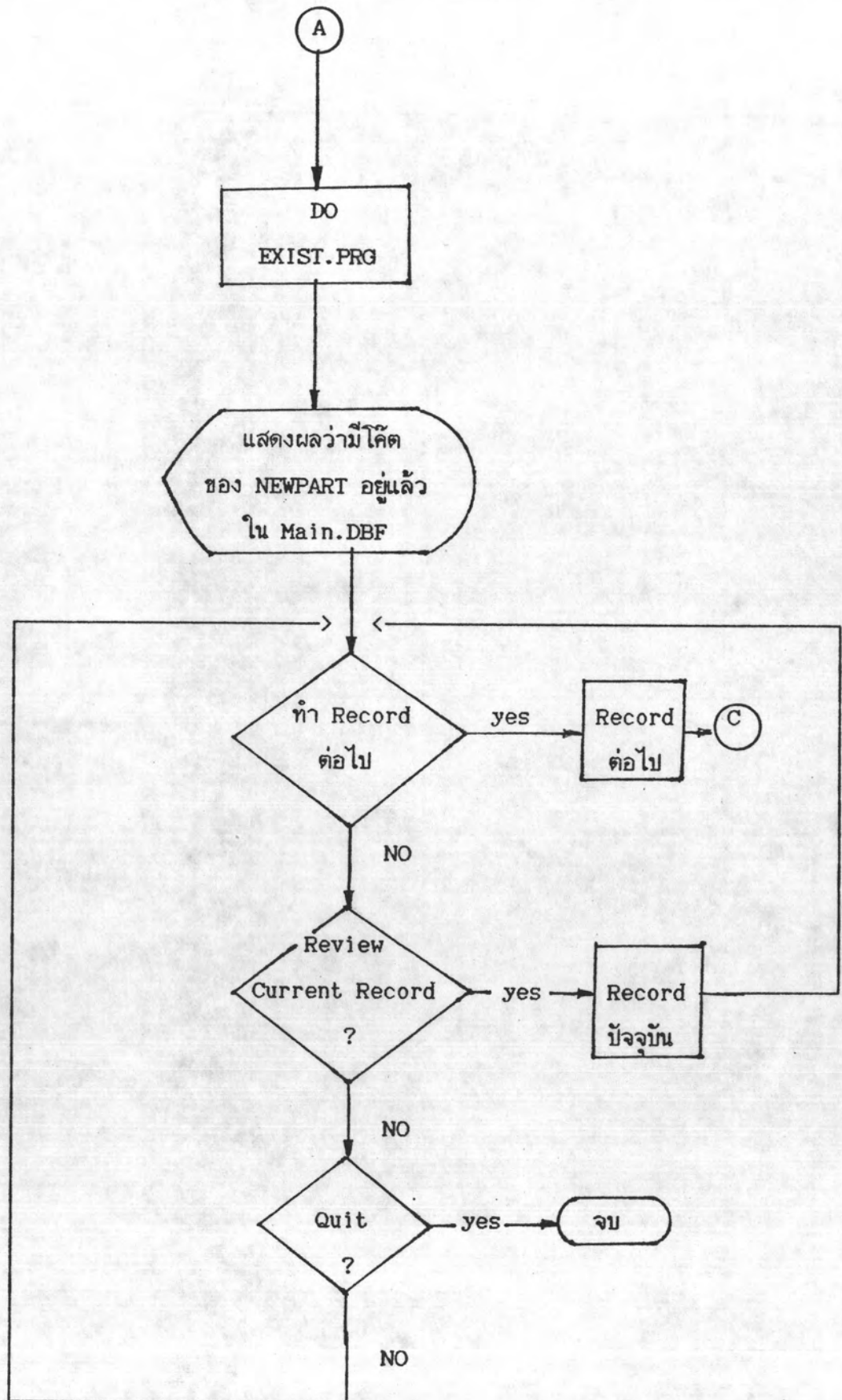


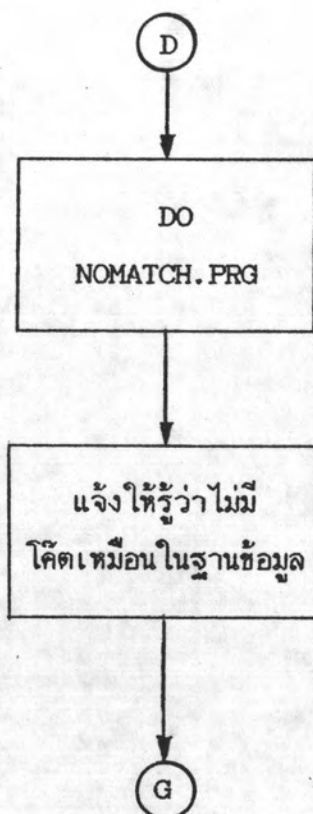


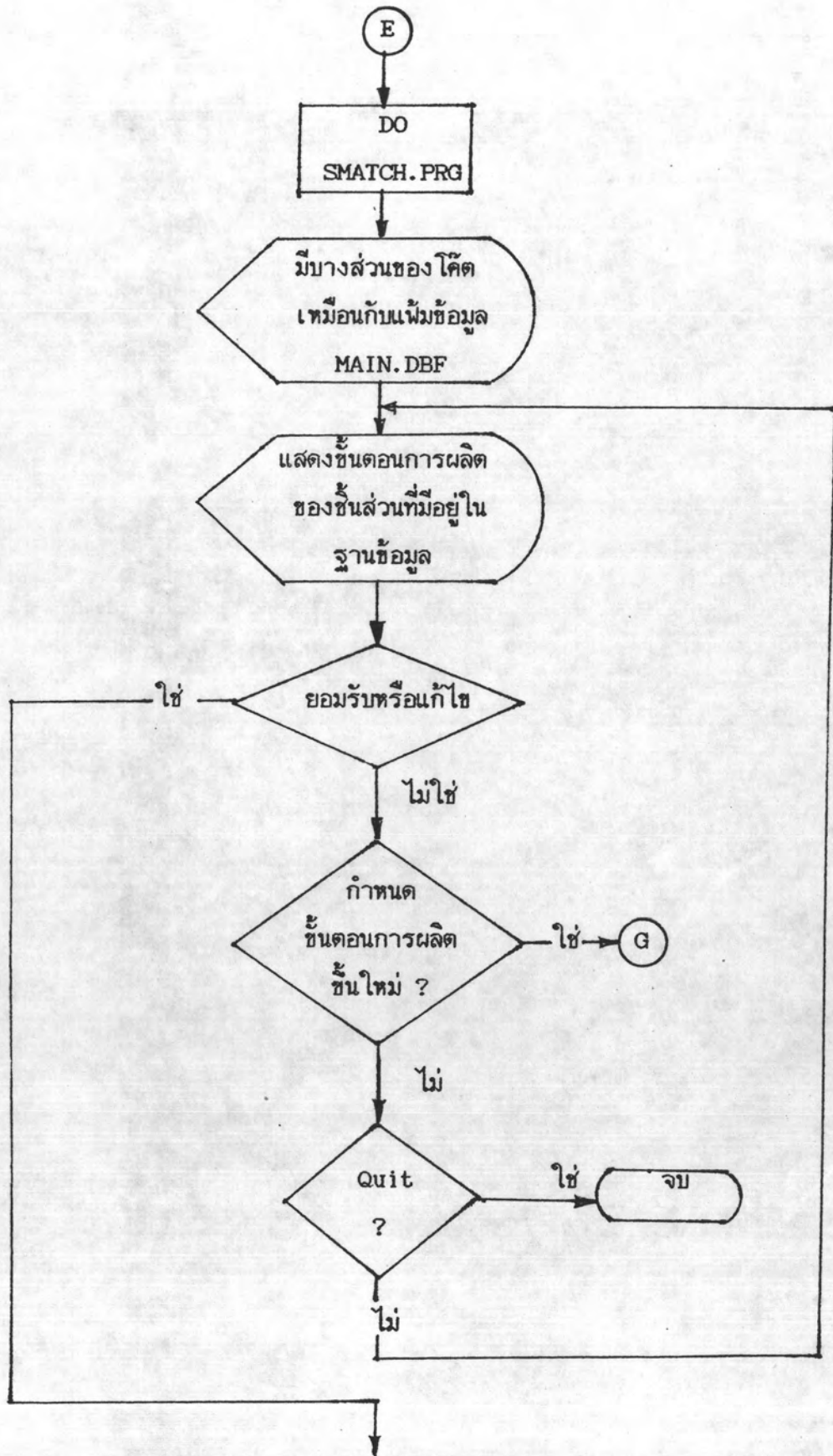


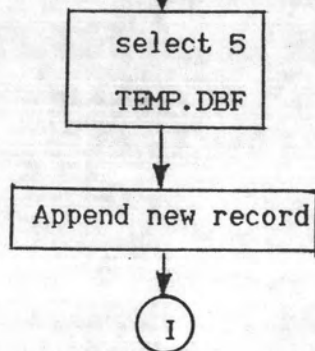
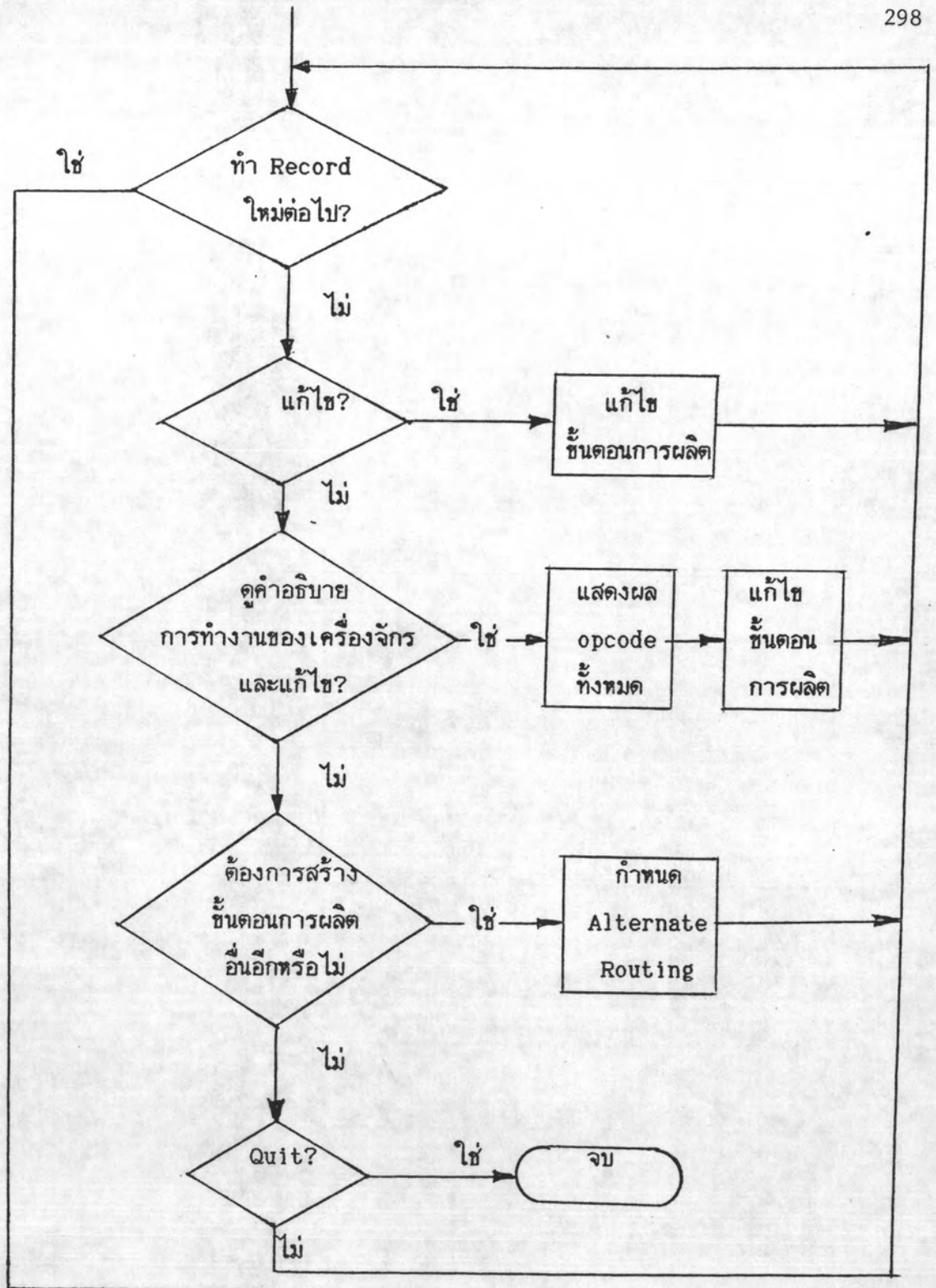


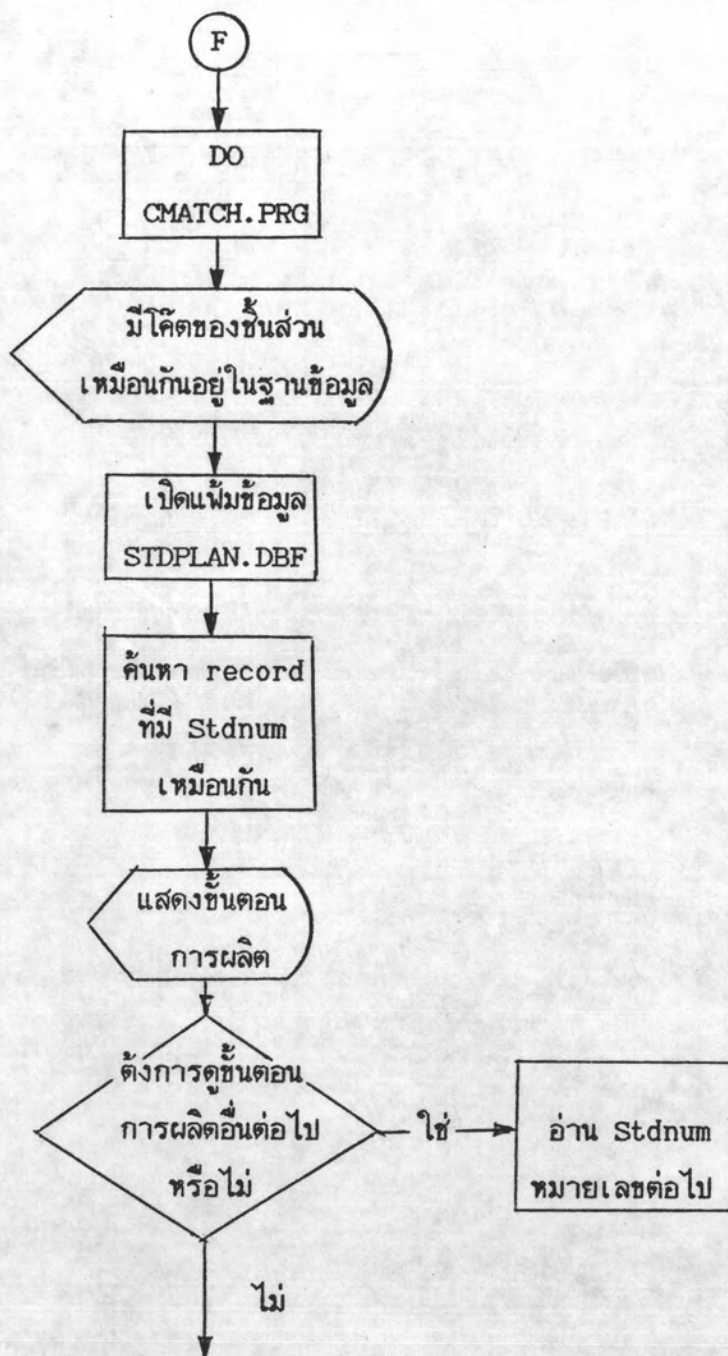


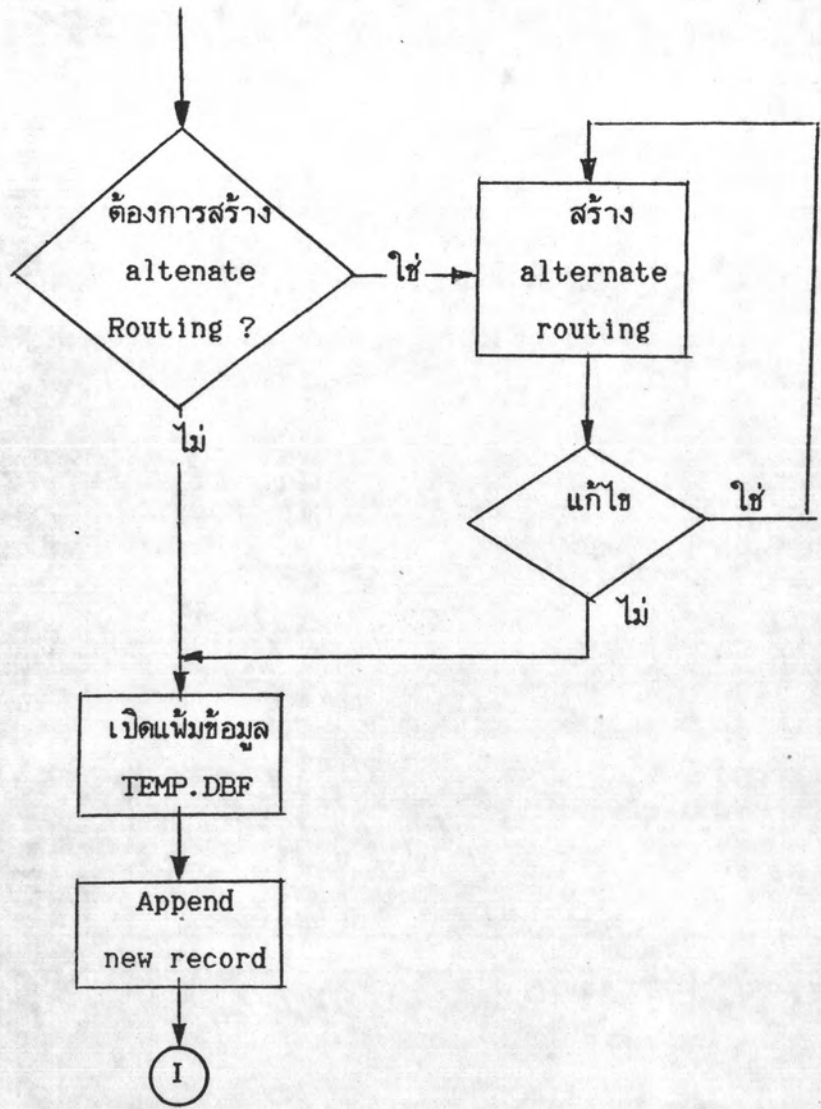


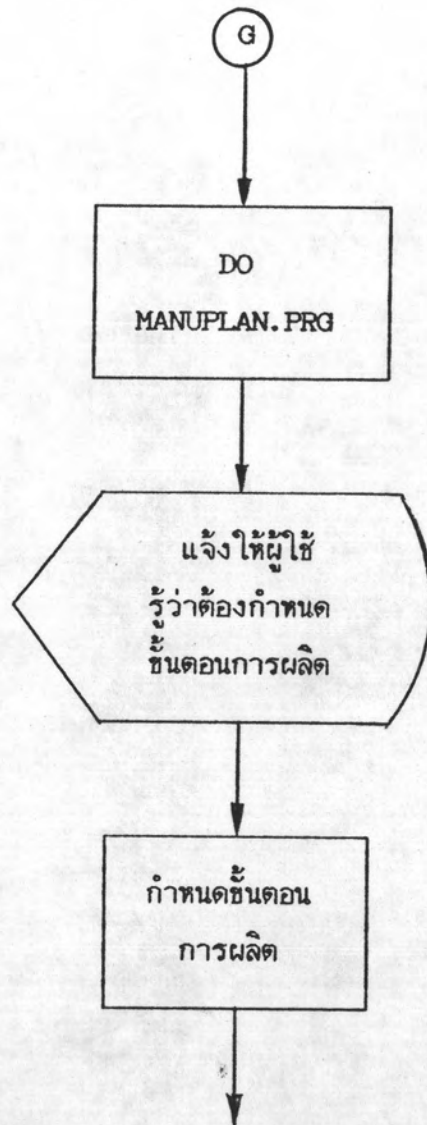




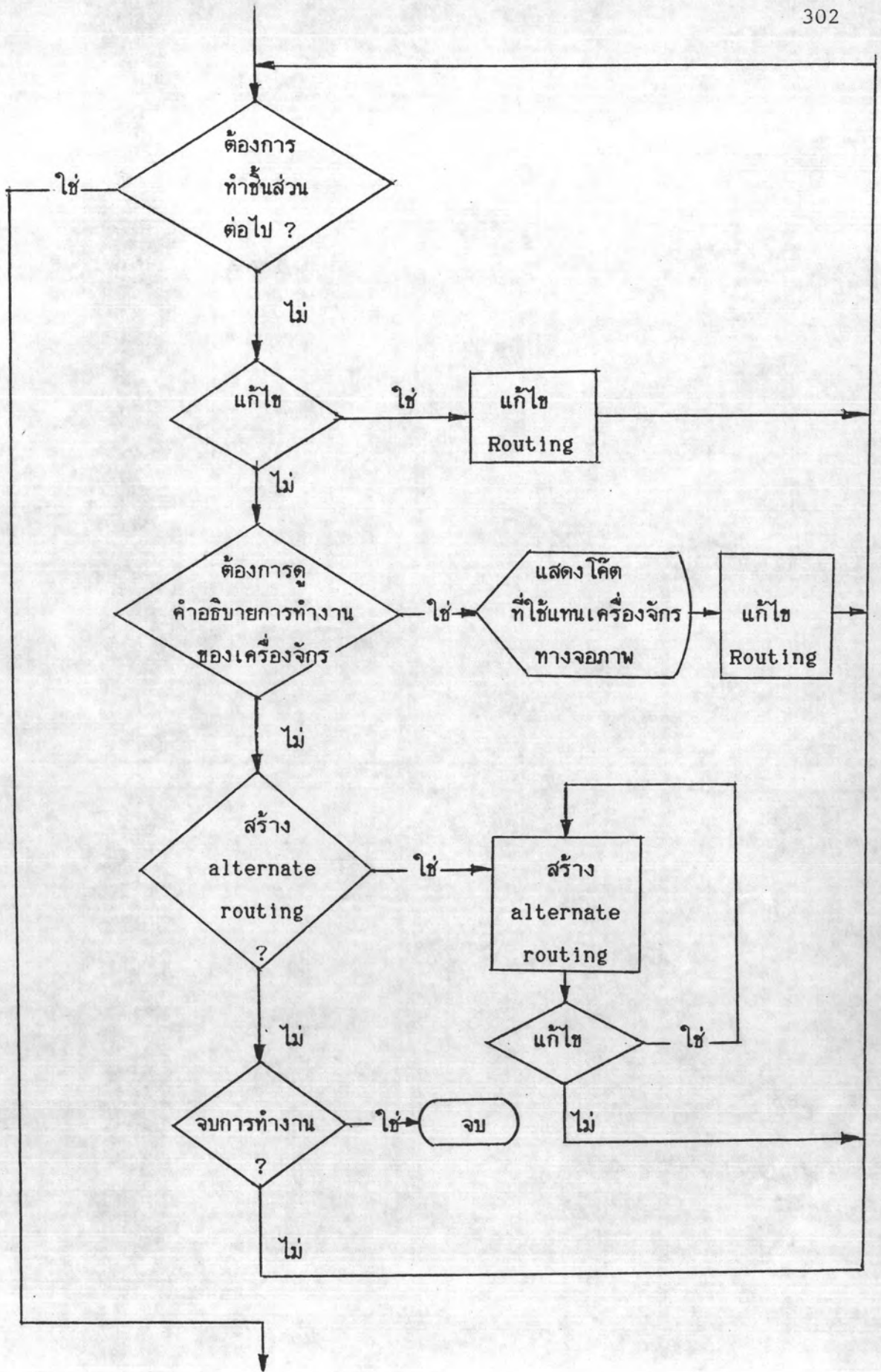


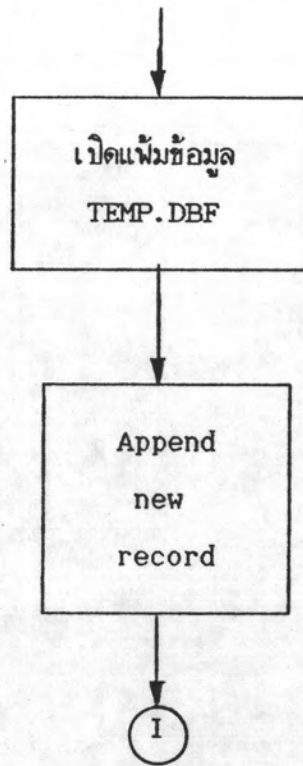


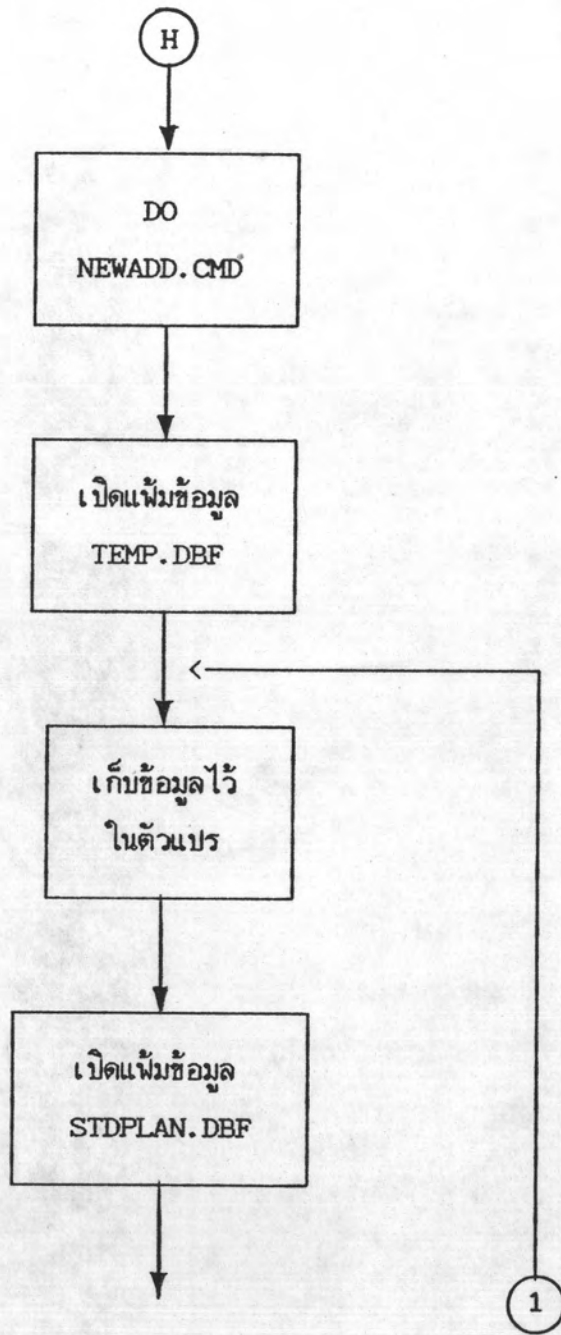


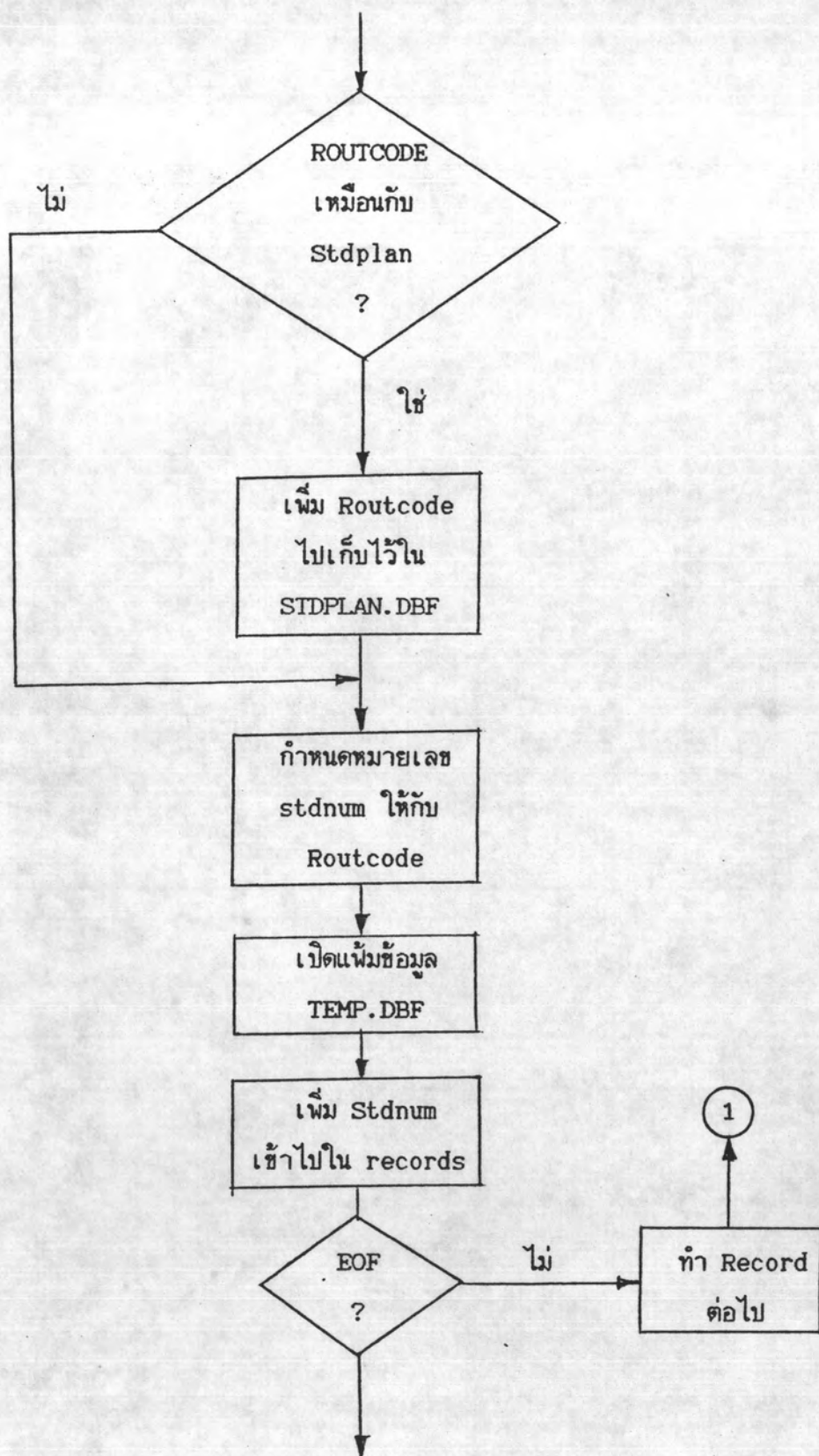


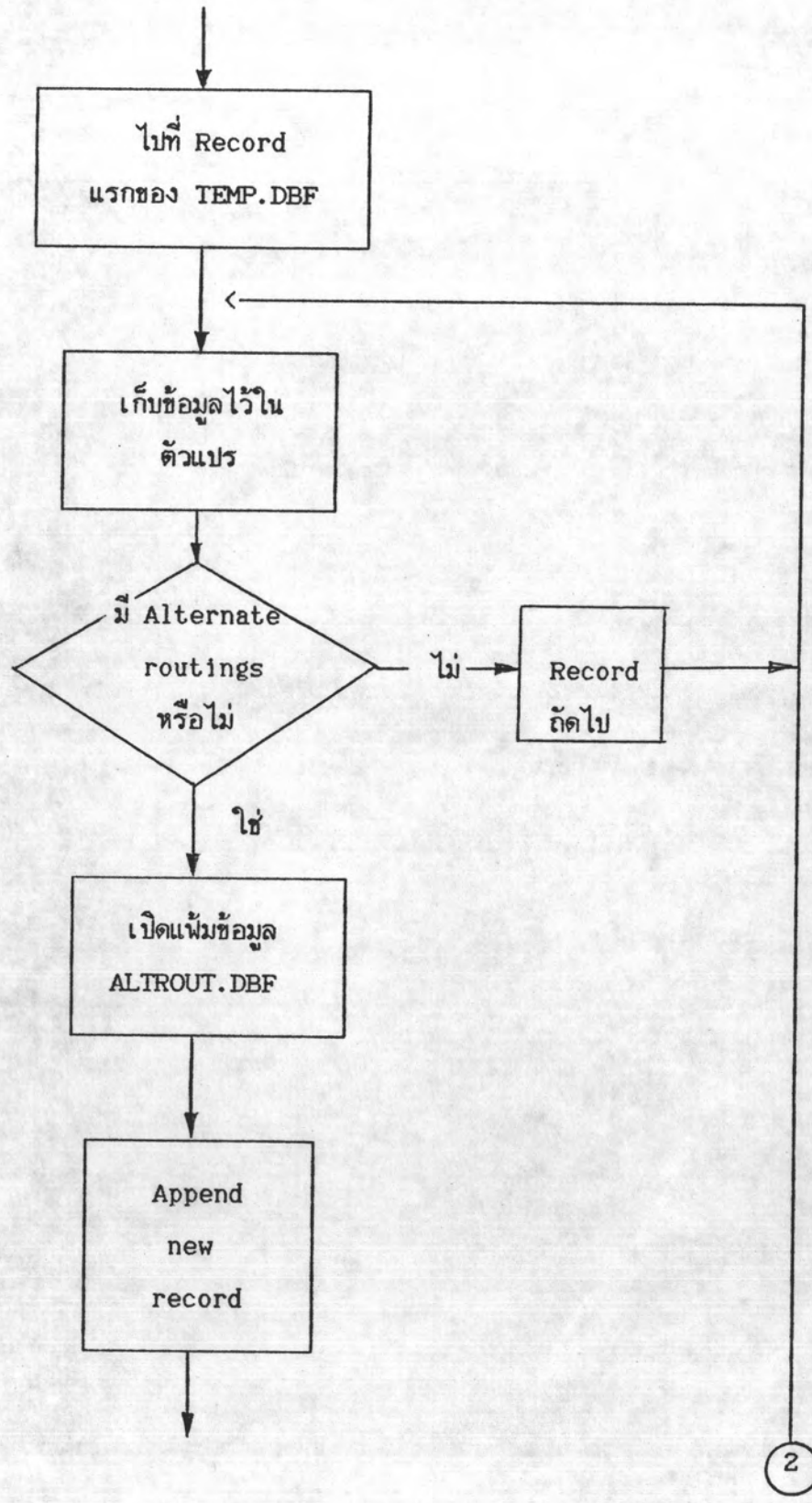


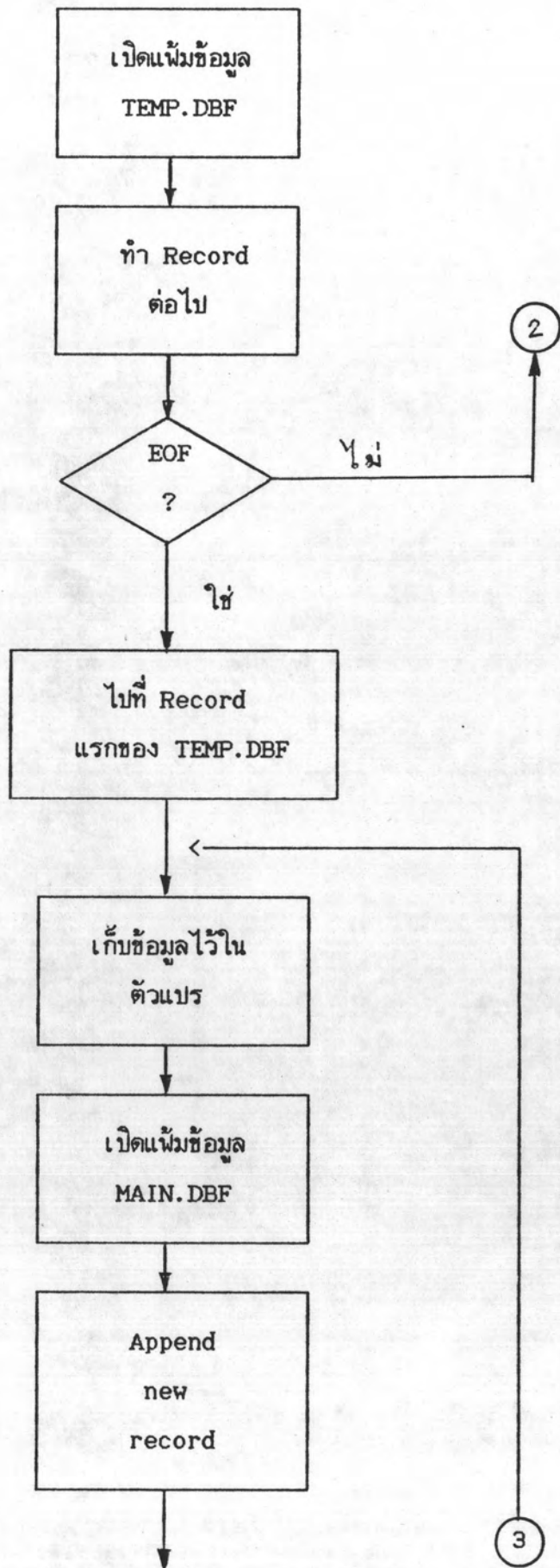


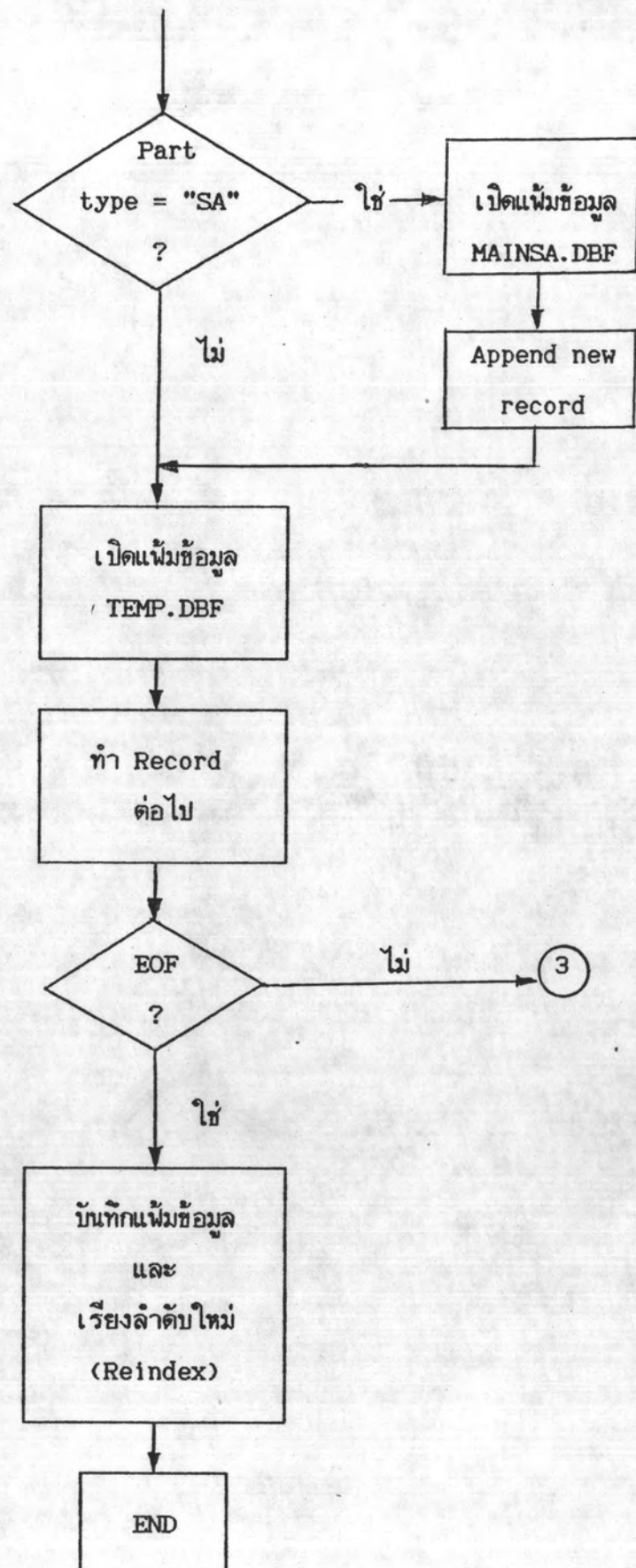












ภาคผนวก จ.

โปรแกรมคอมพิวเตอร์ช่วยในการวางแผนการผลิต



B:THESIS.PRG

Date: 5 January 1989

Time: 00:11:07 P-1

```

1 : SET TALK OFF
2 :
3 :
4 : SET ECHO OFF
5 : CLEA ALL
6 : CLEA
7 :
8 : line = chr(205)+chr(205)+chr(205)+chr(205)+chr(205)+;
9 : chr(205)+chr(205)+chr(205)+chr(205)+chr(205)
10 : @ 7,14 say chr(201)+line+line+line+line+line+chr(187)
11 : @ 21,14 say chr(200)+line+line+line+line+line+chr(188)
12 :
13 : col = 8
14 : DO WHILE col < 21
15 :   @ col,14 say chr(186)
16 :   @ col,65 say chr(186)
17 :   col = col+1
18 : ENDDO
19 :
20 : li = chr(196)+chr(196)+chr(196)+chr(196)+chr(196)+;
21 : chr(196)+chr(196)+chr(196)+chr(196)+chr(196)
22 : sub_li = chr(196)+chr(196)+chr(196)+chr(196)+chr(196)+;
23 : chr(196)+chr(196)+chr(196)
24 : @ 10,14 say chr(199)+li+li+li+li+li+chr(182)
25 : @ 17,14 say chr(199)+li+li+li+li+li+chr(182)
26 :
27 : SET COLO TO /W
28 : @ 3,30 SAY [ :-----: ]
29 : @ 4,30 SAY [ :   THESIS   : ]
30 : @ 5,30 SAY [ :-----: ]
31 : SET COLO TO W
32 : @ 8,27 SAY [Master Industrial Engineer.]
33 : @ 9,28 SAY [Chulalongkorn University.]
34 : @ 12,19 SAY [Computer Asisted Process Planning Software.]
35 : SET COLO TO W+
36 : @ 14,37 SAY [C A P P]
37 : SET COLO TO W
38 : @ 18,17 SAY [Adviser      : Assoc.Prof. Vanchai Rijiravanich ]
39 : @ 19,17 SAY [Co-Adviser : Assoc.Prof. Charoon Mahittafongkul ]
40 : @ 20,17 SAY [Created by : Mr.Somchai Sangaunsak]
41 :
42 :

```

B:THESIS.PRG

Date: 5 January 1989

Time: 00:11:19 P-2

```

43 : @ 22,0
44 : WAIT SPACE(27)+[Press any key to continue...]
45 :
46 : clea
47 : @ 3,18 say [*****]
48 : @ 4,18 say [** PART CLASSIFICATION AND CODING SYSTEM **]
49 : @ 5,18 say [**          FOR FURNITURE INDUSTRY          **]
50 : @ 6,18 say [*****]
51 : *@ 8,3 to 17,77 double
52 : @ 8,3 say chr(201)+line+line+line+line+line+line+line+chr(205)+chr(205)+chr(187)
53 : @ 17,3 say chr(200)+line+line+line+line+line+line+line+chr(205)+chr(205)+chr(188)
54 :
55 : col = 9
56 : DO WHILE col < 17
57 :     @ col,3 say chr(186)
58 :     @ col,76 say chr(186)
59 :     col = col+1
60 : ENDDO
61 :
62 : @ 9,6 say [ The CAPP software will use the part codes to interrogate the]
63 : @ 10,6 say [database to find the appropriate part family and associated standard]
64 : @ 11,6 say [process plan. In this way, the routing of parts for new product]
65 : @ 12,6 say [introductions can be conducted rapidly and accurately. Also, the ]
66 : @ 13,6 say [quality of the routing sequences should improve since most new part]
67 : @ 14,6 say [routings will be based on a retrieved standard process plan. These]
68 : @ 15,6 say [standard plans are tried and tested routingd that have been proven]
69 : @ 16,6 say [to be efficient ways of producing the part.]
70 : @ 20,0
71 : WAIT SPACE(27)+[Press any key to continue...]
72 : do newpart
73 : do codesrch
74 : RETURN

```

D:PARTCODE.PRG

Date: 25 January 1989 Time: 02:15:45 P-1

```

-----
1 | *-----
2 | * PARTCODE.PRG
3 | *-----
4 | * THIS PROGRAM FACILITATES DATA ENTRY OF PART CODES AND COMPANY
5 | * DEPENDENT SUITE AND ITEM NUMBERS INTO THE FILE PARTCODE.DBF
6 | * PREVENT DBASE FROM SENDING EXTRANEIOUS COMMENTS TO THE SCREEN
7 | *-----
8 |
9 | SET TALK OFF
10 | CLEA
11 | clea all
12 | SET PROCEDURE TO SUB_PROG
13 | * --- SUB_PROG.PRG = SACODE.PRG
14 |
15 | TEXT
16 | *****
17 | *****          PART CLASSIFICATION AND CODING          *****
18 | *****          PROCEDURE TO ASSIGN PART CODES          *****
19 | *****          FOR NEW PARTS          *****
20 | *****
21 | *      THIS PROGRAM ALLOWS NEW PARTS AND THEIR CODES INTO THE DATABASE      *
22 | *      FILE PARTCODE.DBF. ALL PARTCODES SHOULD HAVE BEEN DETERMINED      *
23 | *      PRIOR TO THE APPROPRIATE PRINTED TAXONOMIES.      *
24 | *****
25 | *      PARTCODE.DBF (57 BYTES/RECORD) CONTAINS THESE FIELDS      *
26 | *          1) DATE - DATE OF CODE ENTRY          (C,8)          *
27 | *          2) PNUM - PART NUMBER          (C,8)          *
28 | *          3) PLT - PLANT NUMBER          (C,2)          *
29 | *          4) RDIM - ROUGH DIMENSIONS CODE          (C,3)          *
30 | *          5) FDIM - FINISH DIMENSIONS CODE          (C,3)          *
31 | *          6) PTYP - PART TPYE CODE          (C,2)          *
32 | *          7) MAKE - MAKE CODE          (C,1)          *
33 | *          8) MAT - MATERIAL CODE          (C,5)          *
34 | *          9) GEO - PART GEOMETRY CODE          (C,25)          *
35 | *****
36 | ENDTEXT
37 |
38 | @ 23,0
39 | WAIT SPACE(27) + [PRESS ANY KEY TO CONTINUE...]
40 | CLEA ALL
41 | @ 24,0
42 |

```

B:PARTCODE.PRG

Date: 25 January 1989 Time: 02:15:59 P-2

```

-----
43 : *----- OPEN FILE -----
44 : SELE 8
45 : USE SACODE
46 :
47 : SELE 9
48 : USE PARTCODE
49 : *-----
50 :
51 : GO BOTTOM
52 :
53 : * ENTER THE DATE OF CODE ASSIGNMENT AND PLANT NUMBER
54 :
55 : TODAY_DATE = DATE()
56 : PLANTNUM = SPACE(2)
57 :
58 : * ENTER THE DATE OF CODE ASSIGNMENT AND PLANT NUMBER
59 :
60 : @ 23,5 SAY [ENTER TODAYS DATE (MM/DD/YY) ] GET TODAY_DATE
61 : @ 23,53 SAY [ENTER THE PLANT # ] GET PLANTNUM PICT [99]
62 : READ
63 : ENTRYDATE = DTOC(TODAY_DATE)
64 :
65 : * SET UP A CONTINUOUS LOOP TO ALLOW ENTRY OF PART NUMBERS AND CODES
66 :
67 : DO WHILE .T.
68 :     APPE BLANK
69 :     REPL DATE WITH ENTRYDATE
70 :     REPL PLT WITH PLANTNUM
71 :
72 :     NUM = [0]
73 :
74 :     * FORMAT THE SCREEN FOR ENTRY OF PART NUMBERS AND CODES
75 :
76 :     CLEA
77 :     @ 1,10 SAY [*****]
78 :     @ 2,10 SAY [***** PART CLASSIFICATION AND CODING *****]
79 :     @ 3,10 SAY [***** PROCEDURE FOR NON-INTERACTIVE *****]
80 :     @ 4,10 SAY [***** ASSIGNMENT OF PART CODE NUMBERS *****]
81 :     @ 5,10 SAY [*****]
82 :     @ 7,15 SAY [THE PARTNUMBER CONSISTS OF 3 SECTIONS OF COMPANY]
83 :     @ 8,15 SAY [SPECIFIC INFORMATION]
84 :     @ 10,15 SAY [THE FIRST 3 DIGITS DEFINE THE SUITE NUMBER]

```

B:PARTCODE.PRG

Date: 25 January 1989 Time: 02:16:11 P-3

```

-----
85 : @ 11,15 SAY [THE SECOND 3 DIGITS DEFINE THE ARTICLE NUMBER]
86 : @ 12,15 SAY [THE LAST 2 DIGITS DEFINE THE PART NUMBER]
87 : @ 13,10 SAY [-----]
88 : @ 14,23 SAY [ENTER THE PARTNUMBER ] GET PNUM PICT [999-99-9]
89 : @ 15,10 SAY [-----]
90 : @ 17,10 SAY [THE PARTCODE IS DETERMINED FROM THE APPROPRIATE TAXONOMIES]
91 : @ 18,10 SAY [-----]
92 : @ 19,25 SAY [ENTER THE PART CODE INFORMATION]
93 : @ 20,10 SAY [-----]
94 : @ 21,10 SAY [R.DIM F.DIM MAKE P.TYPE MATL PART GEOMETRY]
95 : @ 23,11 GET RDIM PICT [!]
96 : @ 23,17 GET FDIM PICT [!]
97 : @ 23,24 GET MAKE PICT [9]
98 : @ 23,29 GET PTYP PICT [!!]
99 : @ 23,35 GET MAT PICT [!]
100 : @ 23,47 GET GEO PICT [!]
101 : READ
102 :
103 : * SET UP A LOOP TO ALLOW CONTINUATION, EDITING , OR EXIT
104 :
105 : DO WHILE .NOT. (NUM = [1])
106 :     CLEA
107 :     @ 3,5 SAY [PART NUMBER IS ] GET PNUM
108 :     @ 3,40 SAY [DATE OF ENTRY IS ] GET DATE
109 :     @ 6,5 SAY [R. DIMENSIONS CODE ] GET RDIM
110 :     @ 6,40 SAY [F. DIMENSIONS CODE ] GET FDIM
111 :     @ 9,5 SAY [PART TYPE CODE ] GET PTYP
112 :     @ 9,40 SAY [THE MAKE CODE IS ] GET MAKE
113 :     @ 12,5 SAY [MATERIALS CODE ] GET MAT
114 :     @ 12,40 SAY [THE PLANT # IS ] GET PLT
115 :     @ 15,5 SAY [PART GEOMETRY CODE ] GET GEO
116 :     CLEA GETS
117 :
118 :     @ 18,25 SAY [DO YOU WISH TO :]
119 :     @ 19,20 SAY [1] CONTINUE]
120 :     @ 20,20 SAY [2] EDIT THIS RECORD]
121 :     @ 21,20 SAY [3] QUIT]
122 :     @ 23,25 SAY [ENTER THE NUMBER OF YOUR CHOICE ]
123 :
124 :     SET CONSOLE OFF
125 :     WAIT TO NUM
126 :     SET CONSOLE ON

```

B:PARTCODE.PRG

Date: 25 January 1989 Time: 02:16:25 P-4

```
-----
127 |
128 |     DO CASE
129 |     CASE NUM = [1]
130 |         * ALLOWS NORMAL CONTINUATION OF CODE ENTRY
131 |         * OR CALLS THE SUBASSEMBLY CODING ROUTINE
132 |         * IF THE CURRENT PART IS A SUBASSEMBLY
133 |
134 |         IF PTYP = [SA]
135 |             DO SACODE
136 |         ENDIF
137 |     LOOP
138 |
139 |     CASE NUM = [2]
140 |         * ALLOWS EDITING OF THE CURRENT RECORD
141 |         CLEA
142 |         @ 3,10 SAY [RE-ENTER PART NUMBER ] GET PNUM
143 |         @ 5,10 SAY [RE-ENTER R. DIM CODE ] GET RDIM PICT [!]
144 |         @ 7,10 SAY [RE-ENTER P. DIM CODE ] GET FDIM PICT [!]
145 |         @ 9,10 SAY [RE-ENTER PART TYPE CODE ] GET PTYP PICT [!]
146 |         @ 11,10 SAY [RE-ENTER MAKE CODE ] GET MAKE PICT [9]
147 |         @ 13,10 SAY [RE-ENTER MATL. CODE ] GET MAT PICT [!]
148 |         @ 15,10 SAY [RE-ENTER PART GEO. CODE ] GET GEO PICT [!]
149 |         READ
150 |
151 |         IF PTYP = [SA]
152 |             DO SACODE
153 |         ENDIF
154 |     LOOP
155 |
156 |     CASE NUM = [3]
157 |         CLEA ALL
158 |         CLEA
159 |         @ 23,0 SAY [PROGRAM TERMINATED....]
160 |         RETURN
161 |
162 |     OTHERWISE
163 |         CLEA
164 |         @ 10,23 SAY [YOU MUST ENTER A 1, 2 OR 3]
165 |         @ 13,23 SAY [THIS TIME FOLLOW DIRECTIONS]
166 |         @ 16,20 SAY [HIT ANY KEY TO RETURN OPTION MENU]
167 |         SET CONSOLE OFF
168 |         WAIT
```

B:PARTCODE.PRG

Date: 25 January 1989 Time: 02:16:38 P-5

```
-----  
169 |          SET CONSOLE ON  
170 |          LOOP  
171 |          ENDCASE  
172 |          ENDDO  
173 |          *----- END OF DO WHILE .NOT. (NUM = 1) -----  
174 |  
175 |          ENDDO  
176 |          *----- END OF MAIN DO --> DO WHILE .T. -----  
177 |  
178 |          RELEASE ALL  
179 |          SET TALK ON  
180 |          SET PROCEDURE TO  
181 |          CLOSE PROCEDURE  
182 |          RETURN  
-----
```

B:LINK.PRG

Date: 23 January 1989

Time: 00:38:21 P-1

```
-----
1 : * LINK.PRG 13/10/88
2 : * -----
3 : * THIS PROGRAM LINKS PART CODES WITH THE STANDARD PROCESS
4 : * PLAN THAT APPLIES TO THE PART.
5 : * -----
6 :
7 : CLEA ALL
8 : SET ECHO ON
9 : *SET TALK OFF
10 :
11 : *----- OPEN FILE -----
12 : SELECT 1
13 : USE PARTCODE
14 : SELECT 2
15 : USE STDPLAN
16 : *-----
17 :
18 : * COMBINE THE DESIRED FIELDS INTO LINK.DBF
19 :
20 : *-----
21 : SET SAFE OFF
22 : JOIN WITH PARTCODE TO LINK FOR A->PNUM = B->PNUM ;
23 : FIELDS PNUM,STDNUM,DATE,RDIM,FDIM,PTYP,MAKE,MAT,GEO
24 : SET SAFE ON
25 : CLEAR ALL
26 : USE
27 : *-----
28 :
29 : * OPEN THE FILES THAT WILL ALLOW THE LINKING OF THE PARTS THAT
30 : * ARE OF THE SUB-ASSEMBLY TYPE AND HAVE A SACODE ATTACHED TO THEM.
31 : *-----
32 : SELE 2
33 : USE SACODE
34 : SELE 1
35 : USE LINK
36 : *-----
37 :
38 : * COMBINE THE DESIRED FIELDS INTO LINKSA.DBF
39 : *-----
40 : SET SAFE OFF
41 : JOIN WITH SACODE TO LINKSA FOR B->PNUM = A->PNUM FIELD PNUM,PTYP,SACODE
42 : SET SAFE ON
```



B:LINK.PRG

Date: 23 January 1989

Time: 00:38:33 P-2

---

43 : CLEAR ALL  
44 : \*-----  
45 :  
46 : \* INDEX LINK.DBF ON THE PART GEOMETRY AND FORM FEATURE CODE  
47 : \* TO THE INDEX FILE LINKGED  
48 : USE LINK  
49 : SET SAFE OFF  
50 : INDEX ON GEO TO LINKGEO  
51 : SET SAFE ON  
52 :  
53 : CLEAR ALL  
54 : SET TALK ON  
55 : RETURN

---

B:STDPLAN.PRG

Date: 25 January 1989

Time: 02:13:33 P-1

```
-----
1 | * -----
2 | * STDPLAN.PRG
3 | * -----
4 | * THIS PROGRAM WILL INTERROGATE ROUTING.DBF AND SEQUENTIALLY
5 | * COMPARE THE ROUTING SEQUENCE AGAINST ALL OTHERS TO IDENTIFY
6 | * ALL PARTS THAT CARRY THE SAME ROUTING. UNIQUE ROUTING
7 | * CODE SEQUENCES WILL BE ESTABLISHED, ASSIGNED A UNIQUE STDPLAN
8 | * NUMBER AND STORED IN THE DBF FILE STDPLAN.DBF
9 | * -----
10 |
11 | *SET TALK OFF
12 | *SET ECHO OFF
13 | SET EXACT ON
14 | CLEA ALL
15 |
16 | * OPEN THE ROUTING.DBF AND CREATE A TEMPORARY COPY OF IT
17 | * CALLED ROUTTEMP.DBF FOR USE WITH THIS PROGRAM.
18 | * ROUTTEMP.DBF WILL BE ERASED BY THIS PROGRAM.
19 |
20 | COPY FILE ROUTING.DBF TO ROUTTEMP.DBF
21 |
22 | * ----- OPEN FILE -----
23 | SELE 2
24 | USE STDPLAN
25 | SELE 1
26 | USE ROUTTEMP
27 | * -----
28 |
29 | *----- SET VARIABLE -----
30 | GO BOTTOM
31 | NUM = RECNO()
32 | SWITCH = [FIRST]
33 | STDPLANNUM = 1
34 | COUNTER = 1
35 | *-----
36 |
37 | GO TOP
38 |
39 | DO WHILE COUNTER < NUM
40 |     GOTO COUNTER
41 |
42 |     IF DELETED()
```

B:STDPLAN.PRG

Date: 25 January 1989

Time: 02:13:45 P-2

```
-----
43 :     COUNTER = COUNTER + 1
44 :     LOOP
45 :     ENDIF
46 :
47 :     A = TRIM(ROUTCODE)
48 :     B = PNUM
49 :     POINT = COUNTER +1
50 :
51 :     IF POINT > NUM
52 :     LOOP
53 :     ENDIF
54 :
55 :     * POSITION INNER LOOP FILE POINTER AT APPROPRIATE RECORD
56 :
57 :     GOTO POINT
58 :
59 :     * SET UP THE INNER LOOP TO FIND ALL MATCHING ROUTING
60 :     * CODE SEQUENCES AND THEIR CORRESPONDING PART NUMBERS.
61 :
62 :     DO WHILE .NOT. EOF()
63 :     IF DELETED()
64 :     SKIP
65 :     LOOP
66 :     ENDIF
67 :
68 :     C = TRIM(ROUTCODE)
69 :     D = PNUM
70 :
71 :     IF LEN(A) <> LEN(C)
72 :     SKIP
73 :     LOOP
74 :     ENDIF
75 :
76 :     IF LEN(A) = LEN(C) .AND. (A <> C)
77 :     SKIP
78 :     LOOP
79 :     ENDIF
80 :
81 :     IF A=C .AND. SWITCH = 'FIRST'
82 :
83 :     SELECT 2
84 :     * USE STDPLAN.DBF
```

B:STDPLAN.PRG

Date: 25 January 1989 Time: 02:13:56 P-3

```
-----
85 :      APPEND BLANK
86 :
87 :      REPLACE PNUM WITH B
88 :      REPLACE STDNUM WITH STR(STDPLANNUM,5)
89 :      REPLACE STDPLAN WITH A
90 :
91 :      APPEND BLANK
92 :      REPLACE PNUM WITH D
93 :      REPLACE STDNUM WITH STR(STDPLANNUM,5)
94 :      REPLACE STDPLAN WITH C
95 :
96 :      SELECT 1
97 :      * USE ROUTTEMP
98 :      REC = RECNO()
99 :      GO REC
100 :     DELETE
101 :
102 :     SWITCH = 'ELSE'
103 :     SKIP
104 :     LOOP
105 :     ENDIF
106 :
107 :     IF A=C .AND. SWITCH = 'ELSE'
108 :
109 :     SELECT 2
110 :     APPEND BLANK
111 :
112 :     REPLACE PNUM WITH D
113 :     REPLACE STDNUM WITH STR(STDPLANNUM,5)
114 :     REPLACE STDPLAN WITH C
115 :
116 :     SELECT 1
117 :     REC = RECNO()
118 :     GO REC
119 :     DELETE
120 :     SKIP
121 :     LOOP
122 :     ENDIF
123 :     ENDDO
124 :     * ----- TERMINATED INNER LOOP -----
125 :
126 :     IF SWITCH = 'FIRST'
```

B:STDPLAN.PRG

Date: 25 January 1989

Time: 02:14:10 P-4

```
-----  
127 :  
128 :     SELECT 2  
129 :     * USE STDPLAN  
130 :     APPEND BLANK  
131 :  
132 :     REPLACE PNUM WITH B  
133 :     REPLACE STDNUM WITH STR(STDPLANNUM,5)  
134 :     REPLACE STDPLAN WITH A  
135 :  
136 :     SELECT 1  
137 :     ENDIF  
138 :  
139 :     STDPLANNUM = (STDPLANNUM +1)  
140 :     SWITCH = 'FIRST'  
141 :  
142 :     GOTO COUNTER  
143 :     DELETE  
144 :     COUNTER = COUNTER +1  
145 :  
146 : ENDDO  
147 : * -----TERMINATE THE OUTER LOOP -----  
148 :  
149 : GO NUM  
150 :  
151 : IF .NOT. DELETED()  
152 :     A = TRIM(ROUTCODE)  
153 :     B = PNUM  
154 :  
155 :     SELECT 2  
156 :     APPEND BLANK  
157 :  
158 :     REPLACE PNUM WITH B  
159 :     REPLACE STDNUM WITH STR(STDPLANNUM,5)  
160 :     REPLACE STDPLAN WITH A  
161 :  
162 :     SELECT 1  
163 :     GO NUM  
164 :     DELETE  
165 : ENDIF  
166 :  
167 : CLEAR ALL  
168 : ERASE ROUTTEMP.DBF
```

B:STDPLAN.PRG

Date: 25 January 1989

Time: 02:14:21 P-5

-----  
169 |

170 | RELEASE ALL

171 | SET EXACT OFF

172 | SET TALK ON

173 | RETURN  
-----

B:MAIN.PRG

Date: 25 January 1989

Time: 02:19:33 P-1

```
-----
1 : * MAIN.PRG 13/10/88
2 : * -----
3 : * THIS PROGRAM LINKS PART NUMBERS, CODES, STD, PLAN NUMBERS,
4 : * AND ROUTCODES INTO THE PRINCIPAL DATA BASE FILE MAIN.DBF
5 :
6 : * WARNING : THE MAIN.DBF MUST BE CREATED PRIOR TO THE EXECUTION
7 : * OF THIS PROGRAM.
8 : * -----
9 : CLEA ALL
10 :
11 : SET ECHO ON
12 : SET TALK ON
13 : SET SAFE OFF
14 :
15 : SELECT 1
16 : USE LINK
17 : SELECT 2
18 : USE ROUTING
19 : JOIN WITH LINK TO MAIN FOR LINK->PNUM = ROUTING->PNUM
20 : CLEAR ALL
21 :
22 : *----- ADD FIELD CODE, GEO, PLT, FLAG, ALT IN MAIN.DBF -----
23 : RENA MAIN.DBF TO MAIN_BAK.DBF
24 : USE MAIN_BAK
25 : COPY STRU EXTENDED TO TEMPLATE
26 : USE
27 :
28 : USE TEMPLATE
29 :
30 : APPE BLANK
31 : REPL FIELD_NAME WITH [CODE], FIELD_TYPE WITH [C], FIELD_LEN WITH 100
32 :
33 : APPE BLANK
34 : REPL FIELD_NAME WITH [PLT], FIELD_TYPE WITH [C], FIELD_LEN WITH 2
35 :
36 : APPE BLANK
37 : REPL FIELD_NAME WITH [FLAG], FIELD_TYPE WITH [C], FIELD_LEN WITH 1
38 :
39 : APPE BLANK
40 : REPL FIELD_NAME WITH [ALT], FIELD_TYPE WITH [C], FIELD_LEN WITH 1
41 : USE
42 : *-----
```

B:MAIN.PRG

Date: 25 January 1989

Time: 02:19:46 P-2

---

```
43 ;
44 ; CREA MAIN FROM TEMPLATE
45 ; APPE FROM MAIN_BAK
46 ;
47 ; GO TOP
48 ; DO WHILE .NOT. EOF()
49 ;   REPLACE CODE WITH (RDIM+FDIM+PTYP+MAKE+MAT+GEO)
50 ;   SKIP
51 ; ENDDO
52 ; SET SAFE OFF
53 ; INDEX ON CODE TO MAINCODE
54 ; INDEX ON PNUM TO MAINPNUM
55 ; SET SAFE ON
56 ; USE
57 ;
58 ; * CREATE THE FILE MAINSA.DBF AND COPY LINKSA.DBF TO MAINSA.DBF
59 ;
60 ; COPY FILE LINKSA.DBF TO MAINSA.DBF
61 ;
62 ; * CLOSE ALL FILE AND RETURN TO DBASE
63 ; CLEA ALL
64 ; ERASE MAIN_BAK.DBF
65 ; ERASE TEMPLATE.DBF
66 ; SET TALK ON
67 ; SET SAFE ON
68 ; RETURN
```

---



B:L\_OPCODE.PRG

Date: 23 January 1989 Time: 00:44:54 P-1

```
-----  
1 | SET TALK OFF  
2 | SELF 6  
3 | * USE OPCODE  
4 | CLEA  
5 | LI = [=====]  
6 | SET COLO TO W+  
7 | @ 2,28 SAY [OPERATION CODE LISTING...]  
8 | @ 3,10 SAY LI+LI  
9 | @ 4,10 SAY [ OPERATION CODE            OPERATION NAME]  
10 | @ 5,10 SAY LI+LI  
11 | SET COLO TO W  
12 | GO TOP  
13 |  
14 | DO WHILE .NOT. EOF()  
15 |     M = 6  
16 |     @ 6,0 CLEA  
17 |     DO WHILE (M < 23) .AND. (.NOT. EOF())  
18 |         @ M,16 SAY OPCODE + SPACE(10) + OPNAME  
19 |         M = M+1  
20 |         SKIP  
21 |         LOOP  
22 |     ENDDO  
23 |     IF EOF()  
24 |         IF M < 23  
25 |             ? SPACE(30)+[*** END OF DATA ***]  
26 |         ENDIF  
27 |         DO WHILE M < 23  
28 |             ?  
29 |             M = M+1  
30 |         ENDDO  
31 |     ENDIF  
32 |     SET COLO TO W+  
33 |     @ 23,10 SAY LI+LI  
34 |     WAIT SPACE(17)+[PRESS ANY KEY TO CONTINUE...OR PRESS Q TO QUIT] TO TEST  
35 |     SET COLO TO W  
36 |     IF UPPER(TEST) = [Q]  
37 |         EXIT  
38 |     ENDIF  
39 | ENDDO  
40 | RETURN  
-----
```

B:SUB\_PROG.PRG

Date: 23 January 1989 Time: 00:45:45 P-1

```

-----
1 : PROCEDURE SACODE
2 : * sacode.prg      13/10/88
* routine from thehna program is called from partcode.prg and is the data
4 : * only the part being coded is a sub-assembly lead item.
5 : * all sub-assembly coded are keep in the file sacode.dbf.
6 :
7 : clear
8 : text
9 : *****
10 : *
11 : *   The part you are coding is a sub-assembly lead item *
12 : *   It requires the entry of a sub-assembly code      *
13 : *
14 : *****
15 : endtext
16 :
17 : *---- OPEN FILE ----
18 : * select 8
19 : * use sacode
20 : *-----
21 :
22 : SELE 8
23 : go bottom
24 : append blank
25 : SACODE = SPACE(19)
26 : @ 12,13 say "ENTER THE SUB-ASSEMBLY CODE " GET SACODE PICT [!]
27 : READ
28 :
29 : REPLACE SACODE WITH UPPER(SACODE)
30 :
31 : * TRANSFER DATA FROM PARTCODE.PRG INTO SACODE.PRG
32 :
33 : REPLACE DATE WITH PARTCODE->DATE
34 : REPLACE PNUM WITH PARTCODE->PNUM
35 : REPLACE PTYP WITH PARTCODE->PTYP
36 :
37 : * DISPLAY THE INFORMATION BEING ENTERED INTO SACODE.DBF
38 :
39 : DO WHILE .T.
40 :   CLEAR
41 :   @ 7,15 SAY "THE PART NUMBER IS " GET PNUM
42 :   @ 9,15 SAY "THE DATE OF ENTRY IS " GET DATE

```

B:SUB\_PROG.PRG

Date: 23 January 1989 Time: 00:45:57 P-2

```
-----
43 | @ 11,15 SAY "THE PART TYPE IS " GET PTYP
44 | @ 13,15 SAY "THE SUB-ASSEMBLY CODE IS " GET SACODE
45 | CLEA GETS
46 |
47 | @ 15,25 SAY "DO YOU WISH TO : "
48 | @ 16,30 SAY "1. CONTINUE CODING "
49 | @ 17,30 SAY "2. EDIT THIS RECODE"
50 | @ 18,30 SAY "3. EXIT"
51 | @ 20,20 SAY "ENTER THE NUMBER OF YOUR CHOICE "
52 |
53 | SET CONSOLE OFF
54 | WAIT TO NUM
55 | SET CONSOLE ON
56 |
57 | * SET UP A CASE STATEMENT TO ALLOW EDITING OR CONTINUATION
58 |
59 | DO CASE
60 | CASE NUM='1'
61 |
62 |     SELECT 9
63 |     RETURN
64 |
65 | CASE NUM='2'
66 |     CLEAR
67 |     @ 10,10 SAY " PART NUMBER " GET PNUM PICT [99999999]
68 |     @ 12,10 SAY " DATE " GET DATE PICT [99/99/99]
69 |     @ 14,10 SAY " PART TYPE " GET PTYP PICT [!:]
70 |     CLEAR GETS
71 |
72 |     @ 16,10 SAY " SUB-ASSEMBLY CODE " GET SACODE PICT [e:]
73 |     READ
74 |
75 |     * WRITE EDITED DATA INTO SACODE.PRG AND RETURN TO DISPLAY
76 |
77 |     REPLACE SACODE WITH UPPER(SACODE)
78 |     LOOP
79 |
80 | CASE NUM='3'
81 |     USE
82 |     CLEAR ALL
83 |     CLEA
84 |     @ 23,0 SAY [PROGRAM TERMINATED...]
```

B:SUB\_PROG.PRG

Date: 23 January 1989 Time: 00:46:10 P-3

---

```
85 :      CANCEL
86 :
87 :      OTHERWISE
88 :      CLEA
89 :      @ 15,20 SAY [YOU MUST ENTER A 1 OR 2]
90 :      @ 18,20 SAY [HIT ANY KEY TO RETURN TO MENU]
91 :      SET CONSOLE OFF
92 :      WAIT
93 :      SET CONSOLE ON
94 :      LOOP
95 :
96 :      ENDCASE
97 : ENDDO
98 : RETURN
```

---

R:NEWPART.PRG

Date: 25 January 1989

Time: 02:22:41 P-1

```

-----
1 | * NEWPART.PRG
2 | * -----
3 | * THIS PROGRAM FACILITATES DATA ENTRY FOR NEW PARTS. PART CODES
4 | * AND COMPANY DEPENDENT PART NUMBERS (SUITE, ARTICLE, & PART) ARE
5 | * ENTERED INTO THE TEMPORARY FILE NEWPART.DBF
6 | * -----
7 |
8 | SET TALK OFF
9 | SET BELL OFF
10 | CLEA
11 | CLEA ALL
12 |
13 | TEXT
14 | *****
15 | ***** PART CLASSIFICATION AND CODING *****
16 | ***** PROCEDURE TO ASSIGN PART CODES *****
17 | ***** FOR NEW PARTS *****
18 | *****
19 | * This program allows new parts and their code into the database *
20 | * file NEWPART.DBF. All partcodes should have been determined *
21 | * prior to the execution of this program by using the appropriate *
22 | * printed taxonomies. *
23 | *-----*
24 | * NEWPART.DBF (76 bytes/record) contains these fields: *
25 | * #1 DATE - date of code entry (C,8) #7 MAKE - make code (C,1) *
26 | * #2 PNUM - part number (C,8) #8 MAT - material code (C,5) *
27 | * #3 PLT - plant number (C,2) #9 GEO - part geometry code (C,25) *
28 | * #4 RDIM - rough dim. code (C,3) #10 SACODE - sub-assembly code (C,19) *
29 | * #5 FDIM - finished dim. code (C,3) #11 ARTICLE - article name (C,18) *
30 | * #6 PTYP - part type code (C,2) #12 PARTNAME - part name (C,20) *
31 | *****
32 | endtext
33 |
34 | CLEA ALL
35 | *----- OPEN FILE -----
36 | USE NEWPART.DBF
37 | *-----
38 |
39 | GO BOTTOM
40 |
41 | * ENTER THE DATE OF CODE ASSIGNMENT AND PLANT NUMBER
42 |

```

B:NEWPART.PRG

Date: 25 January 1989

Time: 02:22:53 P-2

```

43 : TODAY_DATE = DATE()
44 : PLANTNUM = SPACE(2)
45 : @ 21,5 SAY [ENTER TODAYS DATE (MM/DD/YY) ] GET TODAY_DATE
46 : @ 21,50 SAY [ENTER THE PLANT # ] GET PLANTNUM PICT [99]
47 : READ
48 : ENTRYDATE = DTOC(TODAY_DATE)
49 :
50 : CLEA
51 : TEXT
52 : *****
53 : *****      PART CLASSIFICATION AND CODING PROCEDURE FOR      *****
54 : *****      NON-INTERACTIVE ASSIGNMENT OF PART CODE NUMBERS      *****
55 : *****
56 : ENDTXT
57 :
58 : @ 5,8 SAY [THE PARTNUMBER CONSISTS OF COMPANY SPECIFIC INFORMATION. SUITE]
59 : @ 6,8 SAY [NUMBER (3 DIGITS), ARTICLE NUMBER (3 DIGITS), AND PART NUMBER]
60 : @ 7,8 SAY [(2 DIGITS) ARE CONCATENATED TO FORM THE 8 DIGITS PARTNAME]
61 : @ 8,8 SAY [-----]
62 :
63 : DO WHILE .T.
64 : *----- SET VARIABLE -----
65 : V_PNUM = SPACE(8)
66 : V_RDIM = SPACE(3)
67 : V_FDIM = SPACE(3)
68 : V_PTYP = SPACE(2)
69 : V_MAKE = SPACE(1)
70 : V_MAT = SPACE(5)
71 : V_GRO = SPACE(25)
72 : V_SACODE = SPACE(19)
73 : V_ARTICLE = SPACE(18)
74 : V_PARTNAME = SPACE(20)
75 : *-----
76 :
77 : DO WHILE .T.
78 : @ 9,0 CLEA
79 : @ 9,23 SAY [DATE (MM/DD/YY) ] GET ENTRYDATE
80 : @ 10,23 SAY [ PLANT NUMBER ] GET PLANTNUM PICTURE '99'
81 : @ 11,23 SAY [ PARTNUMBER ] GET V_PNUM PICT [#!]
82 : @ 12,23 SAY [ ARTICLE NAME ] GET V_ARTICLE PICT [#!]
83 : @ 13,23 SAY [ PART NAME ] GET V_PARTNAME PICT [#!]
84 : @ 14,8 SAY [-----]

```

B:NEWPART.PRG

Date: 25 January 1989 Time: 02:23:05 P-3

```

-----
85 : @ 15,8 SAY [THE PART CODE IS DETERMINED FROM THE APPROPRIATE TAXONOMIES]
86 : @ 16,25 SAY [ENTER THE PART CODE INFORMATION]
87 : @ 17,8 SAY [-----]
88 : @ 18,8 SAY [R.DIM F.DIM MAKE P.TYPE MAT PART GEOMETRY]
89 : @ 19,8 GET V_RDIM PICT [?!]
90 : @ 19,15 GET V_FDIM PICT [?!]
91 : @ 19,21 GET V_MAKE PICT [9]
92 : @ 19,27 GET V_PTYP PICT [!!]
93 : @ 19,35 GET V_MAT PICT [?!]
94 : @ 19,44 GET V_GEO PICT [?!]
95 : @ 21,8 SAY [-----]
96 : READ
97 : IF V_PTYP = [SA]
98 : @ 20,25 say [SUB-ASSEMBLY CODE ] GET V_SACODE PICT [?!]
99 : READ
100 : ENDIF
101 :
102 : ASK = [ ]
103 : DO WHILE .NOT. (ASK $ [SEQ])
104 : SET COLO TO 7+
105 : @ 22,21 SAY [PRESS KEY --> E : EDIT THIS RECORD DATA]
106 : @ 23,21 SAY [ --> S : SAVE DATA AND CONTINUE]
107 : @ 24,21 SAY [ --> Q : SAVE DATA AND QUIT ]
108 : SET COLO TO 0/0
109 : @ 23,0 GET ASK PICT [!]
110 : READ
111 : SET COLO TO 7/0
112 : ENDDO
113 :
114 : IF ASK = [E]
115 : LOOP
116 : ELSE
117 : IF ASK $ [SQ]
118 : APPE BLANK
119 : REPLACE DATE WITH ENTRYDATE
120 : REPLACE PNUM WITH V_PNUM
121 : REPLACE PLT WITH PLANTNUM
122 : REPLACE RDIM WITH V_RDIM
123 : REPLACE FDIM WITH V_FDIM
124 : REPLACE PTYP WITH V_PTYP
125 : REPLACE MAKE WITH V_MAKE
126 : REPLACE MAT WITH V_MAT

```

B:NEWPART.PRG

Date: 25 January 1989

Time: 02:23:19 P-4

```
-----  
127 |          REPLACE GEO WITH  V_GEO  
128 |          REPLACE ARTICLE WITH V_ARTICLE  
129 |          REPLACE PARTNAME WITH V_PARTNAME  
130 |          IF V_PTYT = [SA]  
131 |              REPLACE SACODE WITH V_SACODE  
132 |          ENDIF  
133 |  
134 |          IF .NOT. (ASK = [q])  
135 |              EXIT  
136 |          ELSE  
137 |              CLEA  
138 |              IF RECNO() < 2  
139 |                  CLEA ALL  
140 |                  @ 23,0 SAY [PROGRAM TERMINATED...]  
141 |                  RETURN  
142 |              ENDIF  
143 |              @ 10,25 SAY [WORKING...SORT ON RDIM TO NEW]  
144 |              SET TALK ON  
145 |              SORT ON RDIM TO NEW  
146 |              SET TALK OFF  
147 |              USE  
148 |  
149 |              ERASE NEWPART.DBF  
150 |              RENAME NEW.DBF TO NEWPART.DBF  
151 |              CLEA  
152 |              @ 23,0 SAY [PROGRAM TERMINATED...]  
153 |              RETURN  
154 |          ENDIF  
155 |      ENDIF  
156 |  ENDIF  
157 | ENDDO  
158 | ENDDO  
-----
```



B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:00:15 P-1

```

-----
1 : * CODESRCH.PRG      14/10/88
2 : CLEAR ALL
3 : CLEAR
4 : SET TALK OFF
5 : SET ECHO OFF
6 : SET BELL OFF
7 :
8 : IF .NOT. ( FILE("NEWPART.DBF").AND. FILE("MAIN.DBF") .AND.
9 :   FILE("MAINSA.DBF") .AND.FILE("ALTROUT.DBF") .AND.;
10 :   FILE("STDPLAN.DBF").AND.FILE("TEMP.DBF")          )
11 :   @ 2,21 SAY [-----]
12 :   @ 3,23 SAY [MUST HAVE <.DBF FILES> AS FOLLOWING]
13 :   @ 4,21 SAY [-----]
14 :   @ 5,33 SAY [1] NEWPART.DBF]
15 :   @ 6,33 SAY [2] MAIN.DBF]
16 :   @ 7,33 SAY [3] MAINSA.DBF]
17 :   @ 8,33 SAY [4] ALTROUT.DBF]
18 :   @ 9,33 SAY [5] TEMP.DBF]
19 :   @ 10,33 SAY [6] STDPLAN.DBF]
20 :   @ 12,0
21 :   WAIT SPACE(28) +[PRESS ANY KEY TO CONTINUE]
22 :   RETURN
23 : ELSE
24 :   IF .NOT. ( FILE("EXISTS.PRG") .AND.;
25 :     FILE("NOMATCH.PRG") .AND.;
26 :     FILE("SMATCH.PRG") .AND. ;
27 :     FILE("NEWADD.PRG")          )
28 :     CLEA
29 :     @ 2,21 SAY [-----]
30 :     @ 3,23 SAY [MUST HAVE <.PRG FILE> AS FOLLOWING]
31 :     @ 4,21 SAY [-----]
32 :     @ 5,25 SAY [1] EXITSTS.PRG]
33 :     @ 6,25 SAY [2] NOMATCH.PRG]
34 :     @ 7,25 SAY [3] SMATCH.PRG]
35 :     @ 8,25 SAY [4] NEWADD.PRG]
36 :     @ 12,0
37 :     WAIT SPACE(28) +[PRESS ANY KEY TO CONTINUE]
38 :     RETURN
39 :   ENDIF
40 : ENDIF
41 :
42 :

```

B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:00:28 P-2

```

-----
43 : TEXT
44 : *****
45 : *                CODESRCH.PRG                *
46 : *                                                    *
47 : *      THIS PROGRAM WILL PROCESS ALL THE PARTS CURRENTLY CONTAINED IN *
48 : *      NEWPART.DBF AND ALLOW THE FOLLOWING OPERATIONS : *
49 : *                                                    *
50 : *          1) SEARCH MAIN.DBF FOR A CODE MATCH AND RETRIEVE *
51 : *          ANY STD. PLAN(S) THAT APPLY TO THAT CODE *
52 : *                                                    *
53 : *          2) ALLOW THE EDITING OF THE STD. PLAN AND ASSIGNMENT *
54 : *          OF THE NEW PART AND PROCESS PLAN TO MAIN.DBF *
55 : *                                                    *
56 : *          3) IF NO MATCH EXISTED, ALLOW THE CREATION OF NEW PROCESS *
57 : *          PLANS AND ASSIGN THEM TO A PART IN TEMP.DBF *
58 : *                                                    *
59 : *          4) ALTERNATE ROUTCODE SEQUENCES MAY BE DEVELOPED AND *
60 : *          ASSIGNED TO NEW PARTS IN TEMP.DBF *
61 : *                                                    *
62 : *          5) NEW PART INFORMATION IS MERGED INTO MAIN & STDPLAN.DBF *
63 : *                                                    *
64 : *****
65 : ENDTEXT
66 : ?
67 : WAIT SPACE(27) + [PRESS ANY KEY TO CONTINUE]
68 : CLEA
69 :
70 : * SET UP A CONDITION THAT MAKES THIS PROCEDURE IGNORE RECORDS IN
71 : * DBF FILE THAT HAVE ALREADY BEEN MARKED FOR DELETION
72 : * SET DELETED ON
73 :
74 : * ----- FILES USE IN THIS PROGRAM -----
75 : SELE 6
76 : USE OPCODE
77 :
78 : SELE 5
79 : USE TEMP
80 :
81 : SELE 4
82 : USE STDPLAN
83 :
84 : SELECT 3

```

B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:00:41 P-3

```
-----
85 : *USE MAINSA INDEX SAPNUM
86 : USE MAINSA INDEX SACODE
87 :
88 : SELECT 2
89 : USE MAIN INDEX MAINPNUM, MAINCODE
90 :
91 : * OPEN NEWPART.DBF WHICH CONTAINS ALL NEW PARTS WAITING TO HAVE A
92 : * ROUTCODE ASSIGNED AND BE ENTERED INTO MAIN.DBF
93 : SELECT 1
94 : USE NEWPART
95 : GO BOTTOM
96 : *-----
97 :
98 : * INITIALIZE SOME MEMVAR'S THAT WILL BE USED OTHER PROCEDURES
99 :
100 : *----- VARIABLE -----
101 : TOTAL_REC = RECNO()
102 : CHOICE = space(5)
103 : LB = space(5)
104 : KK = space(5)
105 : UU = space(5)
106 : *-----
107 :
108 : * SET UP AN OUTER LOOP TO EXTRACT RECORDS FROM NEWPART.DBF
109 :
110 : GO TOP
111 : DO WHILE .NOT. EOF()
112 :   CLEAR
113 :   REC = RECNO()
114 :   @ 6,28 SAY [TOTAL RECORD      = ]+STR(TOTAL_REC,3)
115 :   @ 8,28 SAY [WORKING ON RECORD = ]+STR(REC,3)
116 :   @ 12,28 SAY [PROCEEDING - PLEASE WAIT]
117 :
118 :   * STORE CURRENT RECORD INFORMATION. TO MEMVAR'S'S IF IT HASN'T BEEN DELETED
119 :
120 :   IF DELETED()
121 :     SKIP
122 :     LOOP
123 :   ELSE
124 :     NDATE = DATE
125 :     NPNUM = PNUM
126 :     NPLT = PLT
```

B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:00:54 P-4

```
-----
127 :      NRDIM = RDIM
128 :      NFDIM = FDIM
129 :      NPTYP = PTYP
130 :      NMAKE = MAKE
131 :      NHAT  = MAT
132 :      NGEO  = GEO
133 :      NSACODE = SACODE
134 :      NARTICLE = ARTICLE
135 :      NPARTNAME = PARTNAME
136 :
137 :      * COMBINE CODE ELEMENTS INTO COMPLETE STRINGS AND STORE TO MENVAR*
138 :
139 :      NSHORT   = RDIM+FDIM+PTYP+MAKE+MAT+GEO
140 :
141 :      IF PTYP = [SA]
142 :          NCODE = NSHORT + NSACODE
143 :      ENDIF
144 :
145 :      ENDIF
146 :
147 :      * SET DBASE TO LOOK FOR EXACT MATCHES ONLY
148 :      SET EXACT ON
149 :
150 :      * OPEN MAIN.DBF USING THE INDEX FILE MAINPNUM.NDX.  THUS, MAIN.DBF
151 :      * IS INDEXED ON THE PART NUMBER AND A TEST CAN BE PERFORMED TO SEE
152 :      * IF THE NEW PART HAS ALREADY BEEN ADDED TO MAIN.DBF
153 :
154 :      SELE 2
155 :      SET INDEX TO MAINPNUM
156 :      GO TOP
157 :
158 :      * TEST TO SEE IF THE NEW PART HAS ALREADY BEEN ENTERED INTO MAIN.DBF
159 :
160 :      SEEK NPNUM
161 :      TNUM = PNUM
162 :      IF NPNUM = TNUM
163 :          SELE 1
164 :          * USE NEWPART
165 :
166 :          DO EXISTS
167 :          LOOP
168 :
```

B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:01:07 P-5

```
-----
169 :   ENDIF
170 :
171 :   * OPEN MAIN.DBF INDEXED ON CODE (MAINCODE.NDX)
172 :   * TEST FOR A CODE MATCH W/O SACODE INCLUDED
173 :
174 :   SELE 2
175 :   SET INDEX TO MAINCODE
176 :   SEEK NSHORT
177 :
178 :   * IF NO MATCH CAN BE FOUND AND THE NEW PART DID NOT
179 :   * PREVIOUSLY EXIST IN MAIN.DBF
180 :
181 :   IF EOF() .AND. (NPNUM<>TNUM)
182 :       SNUM = [0]
183 :       Z   = [ ]
184 :       ZZ  = [ ]
185 :
186 :       SELE 1
187 :       * USE NEWPART
188 :
189 :       DO NOMATCH
190 :
191 :       SELECT 1
192 :       GO REC
193 :       DELETE
194 :       SKIP
195 :       LOOP
196 :   ENDIF
197 :
198 :   * STORE CURRENT RECORD INFORMATION. TO MEMVAR'S
199 :
200 :   REC1 = RECNO()
201 :   MPNUM = PNUM
202 :   MSTDNUM = STDNUM
203 :   MRDIM = RDIM
204 :   MFDIM = FDIM
205 :   MPTYP = PTYP
206 :   MMAKE = MAKE
207 :   MMAT = MAT
208 :   MGEO = GEO
209 :   * COMBINE CODE ELEMENTS INTO LONG STRING
210 :   MSHORT = RDIM+FDIM+PTYP+MAKE+MAT+GEO
```

B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:01:19 P-6

```

211 :
212 : * SET UP A CONDITION THAT PREVENTS MULTIPLE ENTRIES OF A
213 : * PARTICULAR PART INTO MAIN.DBF
214 : IF NPNUM <> TNUM
215 :
216 : * IF THE PART IS A SUBASSEMBLY, OPEN MAINSA.DBF AND ADD
217 : * THE SACODE TO THE LONG CODE STRING AND RETURN TO MAIN.DBF
218 : IF MPTYP = [SA]
219 :
220 :     SELE 3
221 :     * USE MAINSA INDEX SACODE
222 :     SEEK NSACODE
223 :     MSACODE = SACODE
224 :
225 :     SELECT 2
226 :     SET INDEX TO MAINCODE
227 :     * USE MAIN INDEX MAINCODE
228 :     GO REC1
229 :     MCODE = MSHORT+MSACODE
230 :     ENDIF
231 :
232 : * FOR PARTS THAT ARE SUBASSEMBLY'S, STORE THE STDNUM FOR
233 : * THE FIRST OCCURRENCE OF A PARTIAL CODE MATCH, i.e., ONE IN
234 : * WHICH NO COMPLETE CODE MATCH EXISTS, BUT EVERYTHING
235 : * EXCEPT THE SACODE'S ARE A MATCH.
236 :
237 : IF MPTYP = [SA]
238 :     IF NCODE <> MCODE
239 :         STORE STDNUM TO SMATCH
240 :
241 :         DO SMATCH
242 :
243 :         SELECT 1
244 :         GO REC
245 :         DELETE
246 :         SKIP
247 :         LOOP
248 :     ENDIF
249 : ENDIF
250 :
251 : * SET THE PARTIAL CODE VARIABLES EQUAL TO THE COMPLETE
252 : * CODE VARIABLES FOR PARTS THAT ARE NOT SUBASSEMBLY'S

```

B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:01:33 P-7

```

253 :
254 :     IF NPTYP <> 'SA'
255 :         NCODE = NSHORT
256 :         MCODE = MSHORT
257 :     ENDIF
258 :
259 :     * COMPLETE CODE MATCHES ARE NOW HANDLED
260 :
261 :     * STORE ALL UNIQUE STDNUM'S TO A LONG STRING CALLED SNUM
262 :     * BY SETTING UP A LOOP THAT WILL CONTINUE AS LONG AS THE
263 :     * NEXT RECORD IN MAIN.DBF HAS THE SAME CODE AS THE NEW
264 :     * PART.DBF RECORD
265 :
266 :     SNUM = SPACE(5)
267 :
268 :     DO WHILE NCODE = MCODE
269 :         A = STDNUM
270 :
271 :         * TEST FOR PREVIOUS STORAGE OF THE CURRENT STDNUM
272 :         IF AT(A,SNUM) = 0
273 :             SNUM = SNUM + A
274 :             SKIP
275 :
276 :             * STORE NEXT RECORD INFORMATION TO MEMVAR'S
277 :             REC1 = RECNO()
278 :             MPNUM = PNUM
279 :             MSTDNUM = STDNUM
280 :             MRDIM = RDIM
281 :             MFDIM = FDIM
282 :             MPTYP = PTYP
283 :             MMAKE = MAKE
284 :             MHAT = MAT
285 :             MGEO = GEO
286 :             MCODE = RDIM+FDIM+PTYP+MAKE+MAT+GEO
287 :
288 :             IF MPTYP = [SA]
289 :                 SELE 3
290 :                 * USE MAINSA INDEX SAPNUM
291 :                 SEEK MPNUM
292 :                 MCODE = SACODE+MCODE
293 :
294 :                 SELECT 2

```

B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:01:47 P-8

```

-----
295 |          SET INDEX TO MAINCODE
296 |          * USE MAIN INDEX MAINCODE
297 |          GO REC1
298 |          ENDIF
299 |          ELSE
300 |          SKIP
301 |          ENDIF
302 |          ENDDO
303 |
304 |          * DETERMINE THE NUMBER OF UNIQUE STDNUM'S IN THE STRING SNUM
305 |
306 |          SPLANS = ((LRN(SNUM)-5)/5)
307 |
308 |          DO CMATCH
309 |
310 |          ENDIF
311 |
312 |          SELE 1
313 |          GO REC
314 |          DELETE
315 |          IF .NOT. EOF()
316 |          SKIP
317 |          ENDIF
318 |          ENDDO
319 |
320 |          * WHEN THE EOF() FOR NEWPART.DBF IS ENCOUNTERED, OFFER THE USER
321 |          * THE OPTION OF QUITTING, DELETING THE FILE NEWPART.DBF, AND/OR
322 |          * EXECUTING NEWADD.PRG, THIS PROCEDURE WILL TAKE THE NEW PART
323 |          * INFORMATION HELD IN THE TEMPORARY FILE TEMP.DBF AND UPDATE THE
324 |          * FILES MAIN.DBF, MAINSA.DBF, STDPLAN.DBF, AND ALTROUT.DBF
325 |
326 |          DO WHILE .T.
327 |          CLEAR
328 |          @ 8,8 SAY [ALL RECORDS CURRENTLY IN THE FILE ]+:
329 |          [NEWPART.DBF HAVE BEEN PROCESSED]
330 |          @ 12,13 SAY [DO YOU WISH TO :]
331 |          @ 14,18 SAY [1. QUIT]
332 |          @ 15,18 SAY [2. QUIT AND ERASE ALL RECORDS IN NEWPART.DBF]
333 |          @ 16,18 SAY [3. QUIT, ERASE NEWPART, & ADD THE NEW PARTS TO THE DATA BASE]
334 |
335 |          UU = [ ]
336 |          DO WHILE .NOT. (UU $ [123])

```



B:CODESRCH.PRG

Date: 25 January 1989 Time: 02:02:00, P-9

---

```
337 :      @ 18,13 SAY [ENTER THE NUMBER OF YOUR SELECTION] GET UU PICT [9]
338 :      READ
339 :      ENDDO
340 :
341 :      DO CASE
342 :      CASE UU = [1]
343 :          CLEAR ALL
344 :          CLEA
345 :          SET DELETED OFF
346 :          @ 23,0 SAY [PROGRAM TERMINATED...]
347 :          RETURN
348 :      CASE UU = [2]
349 :          CLEAR ALL
350 :          CLEA
351 :          USE NEWPART
352 :          SET SAFE OFF
353 :          ZAP
354 :          SET SAFE ON
355 :          USE
356 :          CANCEL
357 :      CASE UU = [3]
358 :          CLEAR ALL
359 :          USE NEWPART
360 :          SET SAFE OFF
361 :          ZAP
362 :          SET SAFE ON
363 :          USE
364 :
365 :          DO NEWADD
366 :
367 :      ENDCASE
368 :      ENDDO
```

---

B:SMATCH.PRG

Date: 25 January 1989

Time: 02:03:32 P-1

```
-----  
1 : * SMATCH.PRG      14/10/88  
2 :  
3 : * THIS PROCEDURE IS CALLED FROM CODESRCH.PRG WHENEVER A PARTIAL  
4 : * CODE MATCH IS FOUND FOR A NEW PART THAT HAS THE PART TYPE  
5 : * SUBASSEMBLY (PTYP = SA) A PARTIAL CODE MATCH IS ONE THAT  
6 : * MATCHES FOR ALL BUT THE SACODE'S. THIS PROCEDURE WILL ALLOW  
7 : * THE RETRIEVING/EDITING OF THE STDPLAN FOR WHICH THE PARTIAL  
8 : * MATCH WAS FOUND. IF THE STDPLAN IS FOUND TO BE UNUSABLE  
9 : * THIS PROCEDURE WILL CALL MANUPLAN.PRG WHICH WILL FACILITATE  
10 : * THE CONSTRUCTION OF A NEW ROUTING SEQUENCE FOR THE PART  
11 :  
12 : SET TALK OFF  
13 :  
14 : * ----- OPEN FILES USING IN THIS PROGRAM -----  
15 : * SELE 6  
16 : * USE OPCODE  
17 : * SELE 5  
18 : * USE TEMP  
19 : * SELE 4  
20 : * USE STDPLAN  
21 : * -----  
22 :  
23 : SELE 4  
24 : LOCATE FOR STDNUM = SMATCH  
25 : Z = STDPLAN  
26 : ZZ = Z  
27 : CHOICE = [2]  
28 : LB = [2]  
29 :  
30 : DO WHILE .T.  
31 :     CLEAR  
32 :     @ 3,5 SAY [STANDARD PLAN # ] GET SMATCH  
33 :     @ 3,28 SAY [IS A POTENTIAL ROUTING FOR PART NUMBER ] GET NPNUM  
34 :     @ 6,10 SAY [THIS STANDARD PLAN IS BASED ON A PARTIAL CODE MATCH SINCE]  
35 :     @ 8,25 SAY [THE SACODE'S DO NOT MATCH]  
36 :     @ 10,25 SAY [STANDARD PLAN NUMBER ] GET SMATCH  
37 :     @ 12,15 GET Z  
38 :     CLEA GETS  
39 :  
40 :     @ 17,20 SAY [DO YOU WISH TO]  
41 :     @ 19,25 SAY [1. EDIT OR SAVE THIS STANDARD PLAN]  
42 :     @ 20,25 SAY [2. CREATE A NEW STANDARD PLAN]
```

B:SMATCH.PRG

Date: 25 January 1989

Time: 02:03:45 P-2

```
-----
43 :   @ 21,25 SAY [3. QUIT]
44 :
45 :   NUM = [ ]
46 :   DO WHILE .NOT. (NUM $ [123])
47 :       @ 23,20 SAY [ENTER THE NUMBER OF YOUR SELECTION ] GET NUM PICT [9]
48 :       READ
49 :   ENDDO
50 :
51 :   DO CASE
52 :       CASE NUM = [3]
53 :           CLEA
54 :           CLEA ALL
55 :           SET DELETED OFF
56 :           @ 23,0 SAY [PROGRAM TERMINATED...]
57 :           CANCEL
58 :
59 :       CASE NUM = [2]
60 :           CLEA
61 :           @ 12,34 SAY [PLEASE WAIT]
62 :
63 :       DO MANUPLAN
64 :
65 :       RETURN
66 :
67 :       CASE NUM = [1]
68 :           DO WHILE .T.
69 :               IF CHOICE = [2] .OR. CHOICE = [3]
70 :                   CLEAR
71 :                   @ 5,10 SAY [MAKE ANY NECESSARY CHANGES TO MAKE THIS ROUTING]
72 :                   @ 6,14 SAY [APPROPRIATE FOR PART NUMBER ] GET NPNUM
73 :                   @ 8,10 SAY [IF AT ANY POINT YOU NEED TO REFERENCE THE LIST OF]
74 :                   @ 9,10 SAY [OPCODES AND THEIR DESCRIPTION, HIT RETURN AND A]
75 :                   @ 10,10 SAY [MENU ALLOWING ACCESS TO THE OPCODE LIST WILL APPEAR]
76 :                   CLEAR GETS
77 :                   @ 14,10 GET Z
78 :                   READ
79 :               ENDIF
80 :
81 :               CLEAR
82 :               @ 6,10 SAY [THE ROUTING SEQUENCE FOR NUMBER ] GET NPNUM
83 :               @ 6,57 SAY [IS : ]
84 :               @ 8,10 GET Z
```

B:SMATCH.PRG

Date: 25 January 1989

Time: 02:03:58 P-3

```
-----
85 |      CLEAR GETS
86 |
87 |      @ 14,20 SAY [DO YOU WISH TO : ]
88 |      @ 16,25 SAY [1. PROCEED TO THE NEXT PART]
89 |      @ 17,25 SAY [2. EDIT THIS ROUTING SEQUENCE]
90 |      @ 18,25 SAY [3. REVIEW THE OPCODE LIST AND EDIT THIS ROUTING]
91 |      @ 19,25 SAY [4. CREATE AND ALTERNATE ROUTING SEQUENCE]
92 |      @ 22,20 SAY [5. QUIT]
93 |
94 |      CHOICE = [ ]
95 |      DO WHILE .NOT. (CHOICE $ [12345])
96 |          @ 22,20 SAY "ENTER THE NUMBER OF YOUR SELECTION " GET CHOICE
97 |          READ
98 |      ENDDO
99 |
100 |     DO CASE
101 |     CASE CHOICE = [5]
102 |         CLEA
103 |         CLEA ALL
104 |         SET DELETED OFF
105 |         @ 23,0 SAY [PROGRAM TERMINATED...]
106 |         CANCEL
107 |
108 |     CASE CHOICE = [1]
109 |
110 |         SELE 5
111 |         * USE TEMP
112 |         APPEND BLANK
113 |         REPLACE PNUM WITH NPNUM
114 |         REPLACE DATE WITH NDATE
115 |         REPLACE ARTICLE WITH NARTICLE
116 |         REPLACE PARTNAME WITH NPARTNAME
117 |         RRPLACE PLT WITH NPLT
118 |         REPLACE CODE WITH UPPER(NSHORT)
119 |         REPLACE ROUTCODE WITH UPPER(Z)
120 |         REPLACE SACODE WITH UPPER(NSACODE)
121 |         RETURN
122 |
123 |     CASE CHOICE = [2]
124 |         SELR 4
125 |         * USE STDPLAN
126 |         LOOP
```

B:SMATCH.PRG

Date: 25 January 1989

Time: 02:04:13 P-4

```
-----
127 |
128 |     CASE CHOICE = [3]
129 |         CLEAR
130 |
131 |         DO L_OPCODE
132 |
133 |         SELE 4
134 |         * USE STDPLAN
135 |         LOOP
136 |
137 |     CASE CHOICE = [4]
138 |         ZZ = Z
139 |         DO WHILE .T.
140 |             IF LB = [2] .OR. LB = [3]
141 |                 CLEAR
142 |                 @ 5,10 SAY [THIS IS THE ALTERNATE ROUTING FOR PART NUMBE
143 |                 @ 5,58 GET NPNUM
144 |                 CLEAR GETS
145 |
146 |                 @ 8,20 GET ZZ
147 |                 @ 14,10 SAY [YOU MAY EDIT THIS ROUTING AS NECESSARY]
148 |                 READ
149 |             ENDIF
150 |
151 |             CLEAR
152 |             @ 5,10 SAY [THIS IS THE ALTERNATE ROUTING SEQUENCE FOR PART
153 |             @ 5,61 GET NPNUM
154 |             @ 8,21 GET ZZ
155 |             CLEAR GETS
156 |
157 |             @ 14,20 SAY [DO YOU WISH TO : ]
158 |             @ 16,25 SAY [1. PROCEED TO THE NEXT NEW PART]
159 |             @ 17,25 SAY [2. EDIT THIS ALTERNATE SEQUENCE]
160 |             @ 18,25 SAY [3. REVIEW THE OPCODE LIST & EDIT THIS SEQUENCE
161 |             @ 19,25 SAY [4. QUIT]
162 |
163 |             LB = [ ]
164 |             DO WHILE .NOT. (LB $ [1234])
165 |                 @ 21,20 SAY [ENTER THE NUMBER OF YOUR SELECTION ] GET LB
166 |                 READ
167 |             ENDDO
168 |
```

B:SMATCH.PRG

Date: 25 January 1989

Time: 02:04:27 P-5

```

-----
169 |          DO CASE
170 |          CASE LB = [1]
171 |
172 |          SELE 5
173 |          * USE TEMP
174 |          APPEND BLANK
175 |          REPLACE PNUM WITH NPNUM
176 |          REPLACE DATE WITH NDATE
177 |          REPLACE ARTICLE WITH NARTICLE
178 |          REPLACE PARTNAME WITH NPARTNAME
179 |          REPLACE PLT WITH NPLT
180 |          REPLACE CODE WITH UPPER(NSHORT)
181 |          REPLACE SACODE WITH UPPER(NSACODE)
182 |          REPLACE ROUTCODE WITH UPPER(Z)
183 |          REPLACE ALTCODE WITH UPPER(ZZ)
184 |          RETURN
185 |
186 |          CASE LB = [2]
187 |          SELE 4
188 |          * USE STDPLAN
189 |          LOOP
190 |
191 |          CASE LB = [3]
192 |          CLEA
193 |
194 |          DO L_OPCODE
195 |
196 |          SELE 4
197 |          * USE STDPLAN
198 |          LOOP
199 |
200 |          CASE LB = [4]
201 |          CLEA
202 |          CLEA ALL
203 |          SET DELETED OFF
204 |          @ 23,0 SAY [PROGRAM TERMINATED...]
205 |          CANCEL
206 |          ENDCASE
207 |          *----- ENDCASE OF LB -----
208 |          ENDDO
209 |          ENDCASE
210 |          *----- ENDCASE OF CHOICE -----

```

B:SMATCH.PRG

Date: 25 January 1989

Time: 02:04:41 P-6

---

211 :        ENDCASE

212 :        \*----- ENDCASE OF NUM -----

213 :        ENDDO

214 :        \*----- END OF SMATCH.PRG -----

---

B:CMATCH.PRG

Date: 25 January 1989

Time: 02:06:38 P-1

```
-----
1 : * CMATCH.PRG      07/11/88
2 : * THE PROCEDURE CODESRCH.PRG HAS STORED THE MATCHING PART STDNUM'S
3 : * TO STRING CALLED SNUM WITH A LENGTH STORED IN SPLANS.
4 :
5 : CLEA
6 : SET TALK OFF
7 :
8 : *---- OPEN FILE -----
9 : * SELE 6
10 : * USE OPCODE
11 : * SELE 4
12 : * USE STDPLAN
13 : *-----
14 :
15 : * SET UP AN INITIAL SCREEN TO DISPLAY THE NUMBER OF POTENTIAL
16 : * STDPLANS THAT ARE ASSOCIATED WITH THIS PART
17 :
18 : STR_SPLANS = STR(SPLANS,3)
19 : @ 10,5 SAY [THERE ARE ] GET STR_SPLANS
20 : @ 10,COL()+1 SAY [STANDARD PLANS ASSOCIATED WITH PART NUMBER ] GET NPNUM
21 : @ 15,17 SAY [HIT ANY KEY TO BEGIN REVIEWING THE STDPLANS]
22 : SET CONSOLE OFF
23 : WAIT
24 : SET CONSOLE ON
25 :
26 : * SET UP A LOOP TO EXTRACT INDIVIDUAL STDNUM;S FROM THE STRING SNUM
27 : * AND MATCH THEM WITH THE THEIR STDPLAN IN STDPLAN.DBF AND DISPLAY
28 : * THE STDPLANS FOR REVIEW/EDITING/ADDING
29 :
30 : PT1 = 1
31 : PT2 = 6
32 :
33 : * THESE MEMVAR'S NEED TO BE INITIALIZED HERE SINCE THEY ARE USED IN
34 : * PROCEDURES THAT ARE CALLED BY THIS PROCEDURE. SINCE THIS PROCEDURE
35 : * IS CALLED FROM CODESRCH AND IN TURN CALLS MANUPALN, THE NESTING IS
36 : * THREE LEVELS DEEP. APPARENTLY, DBASE ONLY ALLOWS MEMVAR'S TO BE PASSED
37 : * TO THE NEXT LOWEST LEVEL AND CANNOT BE PASSED THRU MORE THAN ONE LEVEL
38 :
39 : LB = [0]
40 : Z = [DUMMY]
41 : ZZ = Z
42 :
```



B:CMATCH.PRG

Date: 25 January 1989

Time: 02:06:51 P-2

```

-----
43 : SELE 4
44 : * ----- MAIN DO LOOP -----
45 : DO WHILE PT1 <= SPLANS
46 :   * EXTRACT A 5 CHARACTER SUBSTRING (EACH SUBSTRING IS A STDNUM)
47 :   * AND DISPLAY IT FOR REVIEW/EDITING/ADD OPTION
48 :
49 :   CLEAR
50 :   VIEW = SUBSTR(SNUM,PT2,5)
51 :   LOCATE FOR STDNUM = VIEW
52 :   Z = STDPLAN
53 :
54 :   STR_PT1 = STR(PT1,2)
55 :   @ 5,10 SAY [THIS IS POTENTIAL ROUTING # ] GET STR_PT1
56 :   @ 5,COL() SAY [ FOR PART NUMBER ] GET NPNUM
57 :   @ 8,20 GET Z PICT [#!]
58 :
59 :   @ 14,10 SAY [DO YOU WISH TO :]
60 :   @ 16,15 SAY [1. REVIEW THE POTENTIAL ROUTNG]
61 :   @ 17,15 SAY [2. ASSIGN THIS ROUTING TO PART NUMBER ] GET NPNUM
62 :   @ 18,15 SAY [3. EDIT THIS SEQUENCE AND ASSIGN IT TO PART NUMBER ] GET NPNUM
63 :   @ 19,15 SAY [4. CREATE A NEW ROUTING SEQUENCE]
64 :   CLEA GETS
65 :
66 :   NUM = [ ]
67 :   DO WHILE .NOT. (NUM $ [1234])
68 :     @ 21,10 SAY [ENTER THE NUMBER OF YOUR SELECTION ] GET NUM
69 :     READ
70 :   ENDDO
71 :
72 :   DO CASE
73 :     CASE NUM=[4]
74 :       CLEAR
75 :
76 :       DO MANUPLAN
77 :
78 :       RELEASE LB,NUM,CHOICE
79 :       RETURN
80 :
81 :     CASE NUM = [1]
82 :       PT1 = PT1 + 1
83 :       PT2 = PT2 + 5
84 :       IF PT1 > SPLANS

```

B:CMATCH.PRG

Date: 25 January 1989

Time: 02:07:04 P-3

```
-----
85 :         PT1 = 1
86 :         PT2 = 6
87 :     ENDIF
88 :     LOOP
89 :
90 :     CASE NUM=[2]
91 :         * SET UP A SCREEN TO ALLOW THE CREATION OF AN ALTERNATE ROUTING
92 :         CLEAR
93 :         @ 5,15 SAY [THIS IS THE ROUTING SEQUENCE FOR PART NUMBER ] GET NPNUM
94 :         @ 8,15 GET Z PICT [!]
95 :         CLEAR GETS
96 :         @ 14,15 SAY [DO YOU WISH TO CREATE AN ALTERNATE ROUTING SEQUENCE ]
97 :
98 :         CHOICE = [ ]
99 :         DO WHILE .NOT. (CHOICE $ [YN])
100 :             @ 16,26 SAY [ENTER (Y/N)? ] GET CHOICE PICT [!]
101 :             READ
102 :         ENDDO
103 :
104 :         * IF AN ALTERNATE ROUTING IS TO BE CREATED
105 :         IF CHOICE=[Y]
106 :
107 :             DO WHILE LB <> [1]
108 :                 CLEAR
109 :                 @ 5,14 SAY [THIS IS THE ROUTING SEQUENCE FOR PART NUMBER ] GET NP
110 :                 CLEAR GETS
111 :                 @ 7,14 GET Z PICT [!]
112 :                 @ 13,14 SAY [YOU MAY EDIT THIS PLAN TO CREATE THE ALTERNATE ROUTI
113 :                 @ 15,14 SAY [IF AT ANY POINT YOU NEED TO REFERENCE THE LIST OF ]
114 :                 @ 16,14 SAY [OPCODES, HIT RETURN AND A MENU ALLOWING ACCESS TO ]
115 :                 @ 17,14 SAY [THE OPCODE LIST WILL APPEAR ]
116 :                 READ
117 :
118 :                 ZZ = Z
119 :                 CLEAR
120 :                 @ 5,15 SAY [THE ALTERNATE ROUTING SEQUENCE FOR PART # ] GET NPNUM
121 :                 @ 5,67 SAY [IS : ]
122 :                 @ 8,20 GET ZZ
123 :                 CLEAR GETS
124 :
125 :                 @ 14,20 SAY [DO YOU WISH TO : ]
126 :                 @ 16,25 SAY [1. PROCEED TO THE NEXT NEW PART ]
```

B:CMATCH.PRG

Date: 25 January 1989

Time: 02:07:19 P-4

```
-----  
127 :           @ 17,25 SAY [2. EDIT THIS ROUTING SEQUENCE]  
128 :           @ 18,25 SAY [3. REVIEW THE OPCODE LIST AND EDIT THIS SEQUENCE]  
129 :  
130 :           LB = [ ]  
131 :           DO WHILE .NOT. (LB $ [123])  
132 :             @ 20,20 SAY [ENTER THE NUMBER OF YOUR SELECTION ] GET LB  
133 :             READ  
134 :             ENDDO  
135 :  
136 :           DO CASE  
137 :             CASE LB=[1]  
138 :               LOOP  
139 :             CASE LB=[2]  
140 :               LOOP  
141 :             CASE LB=[3]  
142 :               CLEAR  
143 :  
144 :             DO L_OPCODE  
145 :  
146 :             SELE 4  
147 :               * USE STDPLAN  
148 :               LOOP  
149 :             ENDCASE  
150 :  
151 :           ENDDO  
152 :           *----- END DO LOOP OF LB <> [1] -----  
153 :  
154 :           SELE 5  
155 :             * USE TEMP  
156 :             APPEND BLANK  
157 :             REPLACE PNUM WITH NPNUM  
158 :             REPLACE DATE WITH NDATE  
159 :             REPLACE ARTICLE WITH NARTICLE  
160 :             REPLACE PARTNAME WITH NPARTNAME  
161 :             REPLACE PLT WITH NPLT  
162 :             REPLACE CODE WITH UPPER(NSHORT)  
163 :             IF NPTYP=[SA]  
164 :               REPLACE SACODE WITH UPPER(NSACODE)  
165 :             ENDIF  
166 :             REPLACE ROUTCODE WITH UPPER(Z)  
167 :             REPLACE ALTCODE WITH UPPER(ZZ)  
168 :
```

B:CMATCH.PRG

Date: 25 January 1989

Time: 02:07:32 P-5

```

-----
169 :      RELEASE Z, ZZ, CHOICE, NUM, LB
170 :      RETURN
171 :      ENDIF
172 :
173 :      IF CHOICE = [N]
174 :          CLEAR
175 :
176 :          SELE 5
177 :          * USE TEMP
178 :          APPEND BLANK
179 :          REPLACE PNUM WITH NPNUM
180 :          REPLACE DATE WITH NDATE
181 :          REPLACE ARTICLE WITH NARTICLE
182 :          REPLACE PARTNAME WITH NPARTNAME
183 :          REPLACE PLT WITH NPLT
184 :          REPLACE CODE WITH UPPER(NSHORT)
185 :          IF NPTYP=[SA]
186 :              REPLACE SACODE WITH UPPER(NSACODE)
187 :          ENDIF
188 :          REPLACE ROUTCODE WITH UPPER(Z)
189 :
190 :          RELEASE Z,ZZ,NUM,CHOICE,LB
191 :          RETURN
192 :      ENDIF
193 :      *----- END OF NUM = [2] -----
194 :
195 :      CASE NUM = [3]
196 :
197 :          DO WHILE LB <> [1]
198 :              LB = [0]
199 :              CLEAR
200 :              @ 5,15 SAY [THIS IS THE ROUTING SEQUENCE FOR PART NUMBER ] GET NPNU
201 :              CLEAR GETS
202 :              @ 8,15 GET Z PICT [!]
203 :
204 :              @ 14,14 SAY [YOU MAY EDIT THIS PLAN AS NECESSARY. ]
205 :              @ 15,14 SAY [IF YOU NEED TO REFERENCE THE LIST OF OPCODES,]
206 :              @ 16,14 SAY [HIT RETURN AND MENU ALLOWING ACCESS TO ]
207 :              @ 17,14 SAY [THE OPCODE LIST WILL BE PRESENTED]
208 :              READ
209 :
210 :              CLEA

```

B:CMATCH.PRG

Date: 25 January 1989

Time: 02:07:45 P-6

```

211 |      @ 5,15 SAY [THIS IS THE ROUTING SEQUENCE FOR PART NUMBER ] GET NPNUM
212 |      @ 8,20 GET Z
213 |      CLEAR GETS
214 |
215 |      @ 14,20 SAY [DO YOU WISH TO :]
216 |      @ 16,25 SAY [1. PROCESS TO THE NEXT PART]
217 |      @ 17,25 SAY [2. EDIT THIS ROUTING SEQUENCE]
218 |      @ 18,25 SAY [3. REVIEW THE OPCODE LIST AND EDIT THIS SEQUENCE]
219 |
220 |      LB = [ ]
221 |      DO WHILE .NOT. (LB # [123])
222 |          @ 20,20 SAY [ENTER THE NUMBER OF YOUR SELECTION ] GET LB
223 |          READ
224 |      ENDDO
225 |
226 |      DO CASE
227 |      CASE LB=[1]
228 |          LOOP
229 |      CASE LB=[2]
230 |          LOOP
231 |      CASE LB=[3]
232 |          CLEAR
233 |
234 |          DO L_OPCODE
235 |
236 |          SELE 4
237 |          * USE STDPLAN
238 |      ENDCASE
239 |      ENDDO
240 |      *----- ENDDO OF LB <> 1 ( END OF CASE NUM = [3] ) -----
241 |
242 |      LB = [0]
243 |      CLEAR
244 |      CHOICE = [ ]
245 |      DO WHILE .NOT. (CHOICE # [YN])
246 |          @ 14,12 SAY [DO YOU WISH TO CREATE AND ALTERNATE ROUTING SEQUENCE ?
247 |          @ 14,COL() GET CHOICE PICT [?]
248 |          READ
249 |      ENDDO
250 |
251 |      * IF AN ALTERNATE ROUTING IS TO BE CREATED
252 |      IF CHOICE = [Y]

```

B:CMATCH.PRG

Date: 25 January 1989

Time: 02:07:59 P-7

```
-----
253 |
254 |     DO WHILE LB <> [1]
255 |         CLEAR
256 |         @ 5,14 SAY [THIS IS THE ROUTING SEQUENCE FOR PART NUMBER ] GET NP
257 |         CLEAR GETS
258 |         @ 7,14 GET Z PICT [!]
259 |         @ 13,14 SAY [YOU MAY EDIT THIS PLAN TO CREATE THE ALTERNATE ROUTI
260 |         @ 15,14 SAY [IF AT ANY YOU NEED TO REFERENCE THE LIST OF]
261 |         @ 16,14 SAY [OPCODES. HIT RETURN AND A MENU ALLOWING ACCESS TO]
262 |         @ 17,14 SAY [THE CODE LIST WILL APPEAR]
263 |         READ
264 |
265 |         ZZ = Z
266 |         CLEAR
267 |         @ 5,15 SAY [THE ALTERNATE ROUTING SEQUENCE FOR PART # ] GET NPNUM
268 |         @ 5,67 SAY [IS : ]
269 |         @ 8,15 GET ZZ
270 |         CLEAR GETS
271 |
272 |         @ 14,20 SAY [DO YOU WISH TO:]
273 |         @ 16,25 SAY [1. PROCESS TO THE NEXT PART]
274 |         @ 17,25 SAY [2. EDIT THIS ROUTING SEQUENCE]
275 |         @ 18,25 SAY [3. REVIEW THE OPCODE LIST AND EDIT THIS SEQUENCE]
276 |
277 |         LB = [ ]
278 |         DO WHILE .NOT. (LB $ [123])
279 |             @ 20,20 SAY [ENTER THE NUMBER OF YOUR SELECTION ] GET LB
280 |             READ
281 |             ENDDO
282 |
283 |         DO CASE
284 |             CASE LB=[1]
285 |                 LOOP
286 |
287 |             CASE LB=[2]
288 |                 LOOP
289 |
290 |             CASE LB=[3]
291 |                 CLEAR
292 |
293 |         DO L_OPCODE
294 |
```

B:CHATCH.PRG

Date: 25 January 1989

Time: 02:08:13 P-8

```
-----
295 |          CLEA
296 |          SELE 4
297 |          * USE STDPLAN
298 |          ENDCASE
299 |          ENDDO
300 |          *----- ENDDO OF LB <> 1 ( IN CHOICE = [Y] ) -----
301 |
302 |          CLEAR
303 |          SELE 5
304 |          * USE TEMP
305 |          APPEND BLANK
306 |          REPLACE PNUM WITH NPNUM
307 |          REPLACE DATE WITH NDATE
308 |          REPLACE ARTICLE WITH NARTICLE
309 |          REPLACE PARTNAME WITH NPARTNAME
310 |          REPLACE PLT WITH NPLT
311 |          REPLACE CODE WITH UPPER(NSHORT)
312 |          IF NPTYP=[SA]
313 |              REPLACE SACODE WITH UPPER(NSACODE)
314 |          ENDIF
315 |          REPLACE ROUTCODE WITH UPPER(Z)
316 |          REPLACE ALTCODE WITH UPPER(ZZ)
317 |
318 |          RELEASE LB, NUM, CHOICE, Z, ZZ, PT1, PT2
319 |          RETURN
320 |      ENDIF
321 |
322 |      IF CHOICE = [N]
323 |          CLEAR
324 |          @ 12,20 SAY [PROCEEDING TO THE NEXT NEW PART]
325 |          SELE 5
326 |          * USE TEMP
327 |          APPEND BLANK
328 |          REPLACE PNUM WITH NPNUM
329 |          REPLACE DATE WITH NDATE
330 |          REPLACE ARTICLE WITH NARTICLE
331 |          REPLACE PLT WITH NPLT
332 |          REPLACE CODE WITH UPPER(NSHORT)
333 |          IF NPTYP=[SA]
334 |              REPLACE SACODE WITH UPPER(NSACODE)
335 |          ENDIF
336 |          REPLACE ROUTCODE WITH UPPER(Z)
```

B:CMATCH.PRG

Date: 25 January 1989

Time: 02:08:26 P-9

```
-----  
337 |  
338 |         RELEASE Z, ZZ, NUM, CHOICE, LB  
339 |         RETURN  
340 |     ENDIF  
341 |     *----- END OF CASE NUM = [3] -----  
342 |     ENDCASE  
343 |     *----- ENDCASE OF NUM -----  
344 | ENDDO  
345 | *----- END OF DO WHILE PT1 <= SPLANS -----  
346 | RELEASE Z, ZZ, NUM, CHOICE, LB  
347 | RETURN  
-----
```



B:NOMATCH.PRG

Date: 25 January 1989 , Time: 02:17:18 P-1

---

```
1 : * NOMATCH.PRG 14/10/89
2 :
3 : * THIS PROCEDURE IS CALLED FROM CODESRCH.PRG WHENEVER THE CODE
4 : * FOR A NEW PART BEING PROCESSED IS NOT FOUND TO EXIST IN MAIN.DBF
5 : * CONSEQUENTLY, NO STDPLAN CAN BE RETRIEVED FOR EDITING AND A NEW
6 : * ROUTING SEQUENCE MUST BE BUILT FROM SCRATCH. THIS PROCEDURE CALLS
7 : * MANUPLAN.PRG TO ALLOW THE CONSTRUCTION OF THE ROUTING ARE DETERMINED,
8 : * THE NEW PART INFORMATION IS ENTERED INTO TEMP.DBF AND DELETED FROM
9 : * NEWPART.DBF
10 :
11 : CLEAR
12 : @ 5.14 SAY [NO CODE MATCH FOR PART NUMBER] GET NPNUM
13 : @ 5.54 SAY [IS FOUND]
14 : CLEAR GET
15 : @ 8.19 SAY [A ROUTING SEQUENCE MUST BE CONSTRUCTED]
16 : @ 11.27 SAY [PRESS ANY KEY TO CONTINUE]
17 : SET CONSOLE OFF
18 : WAIT
19 : SET CONSOLE ON
20 :
21 : CLEAR
22 : DO MANUPLAN
23 : RETURN
```

---

B:EXISTS.PRG

Date: 23 January 1989

Time: 00:39:25 P-1

```
-----  
1 : * EXISES.PRG 14/10/88  
2 : * THIS PROCEDURE IS CALLED FROM CODESRCH.PRG WHENEVER A NEW PART  
3 : * IS BEING ENTERED INTO THE FILE MAIN.DBF AND THE NEW PART NUMBER  
4 : * IS FOUND TO ALREADY EXIST IN MAIN.DBF. PART NUMBERS SHOULD BE UNIQUE.  
5 :  
6 : * SET UP A SCREEN  
7 : * CURRENT BASE IS MAIN.DBF INDEX MAINPNUM  
8 : DO WHILE .T.  
9 :     REC1 = RECNO()  
10 :     CLEAR  
11 :     @ 5,5 SAY "NEW PART NUMBER " GET NPNUM  
12 :     @ 5,32 SAY "ALREADY EXISTS IN MAIN.DBF"  
13 :     CLEA GETS  
14 :     @ 7,10 SAY "DO YOU WISH TO :"  
15 :     @ 9,20 SAY "1. PROCEED TO NEXT NEW PART"  
16 :     @ 10,20 SAY "2. REVIEW/EDIT CURRENT RECORD"  
17 :     @ 11,20 SAY "3. QUIT"  
18 :     CHOICE = [ ]  
19 :     DO WHILE .NOT. (CHOICE $ [123])  
20 :         @ 14,10 SAY "ENTER THE NUMBER OF YOUR SELECTION : " GET CHOICE  
21 :         READ  
22 :     ENDDO  
23 :     DO CASE  
24 :         CASE CHOICE = [1]  
25 :             CLEAR  
26 :             SELE 1  
27 :             IF .NOT. EOF()  
28 :                 SKIP  
29 :             ENDIF  
30 :             RETURN  
31 :         CASE CHOICE = [2]  
32 :             CLEAR  
33 :             DO ED_EXIST  
34 :             LOOP  
35 :         CASE CHOICE = [3]  
36 :             CLEAR  
37 :             CLEAR ALL  
38 :             SET DELETED OFF  
39 :             CANCEL  
40 :     ENDCASE  
41 : ENDDO  
-----
```

B:ED\_EXIST.PRG

Date: 25 January 1989 Time: 02:11:45 P-1

```
-----
1 | CLEA
2 | SET TALK OFF
3 |
4 | *----- OPEN FILE -----
5 | * SELE 1
6 | * USE NEWPART
7 | * SELE 2
8 | * USE MAIN INDEX MAINCODE
9 | *-----
10 |
11 | LINE = [-----]
12 | @ 1,0 SAY LINE+LINE
13 | SET COLO TO /W
14 | @ 2,27 SAY [ EDIT DATA IN NEWPART.DBF ]
15 | SET COLO TO W
16 | @ 3,0 SAY LINE+LINE
17 | SET COLO TO W+
18 | @ 4,5 SAY [DATA OF MAIN.DBF DATA OF NEWPART.DBF]
19 | SET COLO TO W
20 |
21 | SELE 2
22 |
23 | @ 5,0 SAY [ UPDATE DATE ] GET DATE
24 | @ 6,0 SAY [ PART NUMBER ] GET PNUM
25 | @ 7,0 SAY [ PLANT NUMBER ] GET PLT
26 | @ 8,0 SAY [ ROUGH DIM ] GET RDIM
27 | @ 9,0 SAY [ FINISHED DIM ] GET FDIM
28 | @ 10,0 SAY [ PART PTYP ] GET PTYP
29 | @ 11,0 SAY [ MAKE CODE ] GET MAKE
30 | @ 12,0 SAY [ MATERIAL ] GET MAT
31 | @ 13,0 SAY [ GEOMETRY ] GET GEO
32 | @ 14,0 SAY [ STANDARD NO# ] GET STDNUM
33 | @ 15,0 SAY [ ARTICLE ] GET ARTICLE
34 | @ 16,0 SAY [ PART NAME ] GET PARTNAME
35 | @ 17,0 SAY [ FLAG ] GET FLAG
36 | @ 18,0 SAY [ ALT ] GET ALT
37 | @ 19,0 SAY [ ROUT CODE ] GET ROUTCODE
38 | @ 21,0 SAY [ CODE ] GET CODE
39 | CLEA GETS
40 | @ 23,0 SAY LINE+LINE
41 |
42 | SELE 1
```

B:ED\_EXIST.PRG

Date: 25 January 1989 Time: 02:11:57 P-2

```
-----  
43 | * USE NEWPART  
44 |  
45 | V_DATE = DATE  
46 | V_PNUM = PNUM  
47 | V_PLT = PLT  
48 | V_RDIM = RDIM  
49 | V_FDIM = FDIM  
50 | V_PTyp = PTyp  
51 | V_MAKE = MAKE  
52 | V_MAT = MAT  
53 | V_GEO = GEO  
54 | V_SACODE = SACODE  
55 | V_ARTICLE = ARTICLE  
56 | V_PARTNAME = PARTNAME  
57 |  
58 | @ 5,40 SAY [ UPDATE DATE ] GET V_DATE  
59 | @ 6,40 SAY [ PART NUMBER ] GET V_PNUM  
60 | @ 7,40 SAY [ PLANT NUMBER ] GET V_PLT  
61 | @ 8,40 SAY [ ROUGH DIM ] GET V_RDIM  
62 | @ 9,40 SAY [ FINISHED DIM ] GET V_FDIM  
63 | @ 10,40 SAY [ PART PTyp ] GET V_PTyp  
64 | @ 11,40 SAY [ MAKE CODE ] GET V_MAKE  
65 | @ 12,40 SAY [ MATERIAL ] GET V_MAT  
66 | @ 13,40 SAY [ GEOMETRY ] GET V_GEO  
67 | @ 14,40 SAY [ SUB-ASSEMBLY ] GET V_SACODE  
68 | @ 15,40 SAY [ ARTICLE ] GET V_ARTICLE  
69 | @ 16,40 SAY [ PART NAME ] GET V_PARTNAME  
70 | @ 17,40 SAY LINE  
71 | READ  
72 |  
73 | ASK = [ ]  
74 | SET COLO TO W+  
75 | DO WHILE .NOT. (ASK $ [yYnN])  
76 | @ 24,18 SAY [ARE YOU SURE TO SAVE THIS NEW DATA ? {Y/N} ] GET ASK  
77 | READ  
78 | ENDDO  
79 | SET COLO TO W  
80 |  
81 | IF ASK $ [Nn]  
82 | SELE 2  
83 | RETURN  
84 | ELSE
```

B:ED\_EXIST.PRG

Date: 25 January 1989 Time: 02:12:08 P-3

---

```
85 | REPL DATE WITH V_DATE
86 | REPL PNUM WITH V_PNUM
87 | REPL PLT WITH V_PLT
88 | REPL RDIM WITH V_RDIM
89 | REPL FDIM WITH V_FDIM
90 | REPL PTYP WITH V_PTYT
91 | REPL MAKE WITH V_MAKE
92 | REPL MAT WITH V_MAT
93 | REPL GEO WITH V_GEO
94 | REPL SACODE WITH V_SACODE
95 | REPL ARTICLE WITH V_ARTICLE
96 | REPL PARTNAME WITH V_PARTNAME
97 | ENDIF
98 | SELE 2
99 | RETURN
```

---

B:MANUPLAN.PRG

Date: 25 January 1989 Time: 02:18:18 P-1

```

-----
1 | *----- MANUPLAN.PRG 10/11/31 -----
2 | * THIS PROCEDURE IS CALLED FROM NOMATCH.PRG, SMATCH.PRG AND
3 | * CMATCH.PRG
4 |
5 | SET TALK OFF
6 |
7 | *---- OPEN FILE -----
8 | * SELE 6
9 | * USE OPCODE
10 | * SELE 5
11 | * USE TEMP
12 | * SELE 4
13 | * USE STDPLAN
14 | * SELE 2
15 | * USE MAIN INDEX MAINCODE
16 | *-----
17 |
18 | *---- INITIAL VARIABLE -----
19 | H = 0
20 | U = SPACE(60)
21 | V = SPACE(60)
22 | W = SPACE(60)
23 | X = SPACE(60)
24 | Y = SPACE(14)
25 | Z = SPACE(100)
26 | ZZ = SPACE(100)
27 | *-----
28 |
29 | * CURRENT BASE = MAIN.DBF INDEX MAINCODE AT SELECT AREA 2
30 |
31 | DO WHILE .T.
32 |     CLEAR
33 |
34 |     IF H = 0
35 |         @ 5,10 SAY "A ROUTING SEQUENCE MUST BE CONSTRUCTED FOR PART NUMBER "
36 |         @ 5,COL() GET NPNUM
37 |         CLEAR GETS
38 |     ENDIF
39 |
40 |     IF H = 1
41 |         @ 5,10 SAY "THIS IS TO BE AN ALTERNATE SEQUENCE FOR PART NUMBER "
42 |         @ 5,COL() GET NPNUM

```

B:MANUPLAN.PRG

Date: 25 January 1989 Time: 02:18:30 P-2

```
-----
43 |      CLEAR GETS
44 |      ENDIF
45 |
46 |      @ 8,19 SAY "ENTER THE ROUTING OPCODES SEQUENTIALLY"
47 |      @ 10,15 SAY "IF AT ANY POINT YOU NEED TO REFERENCE THE LIST OF"
48 |      @ 11,15 SAY "OPCODES AND THEIR DESCRIPTIONS, HIT RETURN AND A "
49 |      @ 12,15 SAY "MENU ALLOWING ACCESS TO THE OPCODE LIST WILL APPEAR"
50 |
51 |      IF H=0
52 |          @ 14,10 GET Z pict [!]
53 |      ENDIF
54 |
55 |      IF H=1
56 |          @ 14,10 GET ZZ pict [!]
57 |      ENDIF
58 |
59 |      READ
60 |
61 |      CLEAR
62 |      IF H=0
63 |          @ 6,10 SAY "THE ROUTING SEQUENCE FOR PART NUMBER " GET NPNUM
64 |          @ 6,COL() SAY " IS : "
65 |          @ 8,10 GET Z pict [!]
66 |      ENDIF
67 |
68 |      IF H=1
69 |          @ 6,10 SAY "THE ALTERNATE ROUTING SEQUENCE FOR PART NUMBER " GET NPNUM
70 |          @ 6,COL() SAY " IS: "
71 |          @ 8,10 GET ZZ pict [!]
72 |      ENDIF
73 |
74 |      CLEA GETS
75 |
76 |      @ 14,10 SAY "DO YOU WISH TO: "
77 |      @ 16,15 SAY "1. PROCEED TO THE NEXT NEW PART"
78 |      @ 17,15 SAY "2. EDIT THIS ROUTING SEQUENCE"
79 |      @ 18,15 SAY "3. REVIEW THE OPCODE LIST AND EDIT THIS ROUTING"
80 |
81 |      IF H=0
82 |          @ 19,15 SAY "4. CREATE AN ALTERNATE ROUTING SEQUENCE"
83 |          @ 20,15 SAY "5. QUIT"
84 |          @ 22,10 SAY "ENTER THE NUMBER OF YOUR SELECTION"
```

B:MANUPLAN.PRG

Date: 25 January 1989 Time: 02:18:42 P-3

```
-----
85 : ELSE
86 :     @ 19,15 SAY "4. QUIT"
87 :     @ 21,10 SAY "ENTER THE NUMBER OF YOUR SELECTION"
88 : ENDIF
89 :
90 : SET CONSOLE OFF
91 : WAIT TO NUM
92 : SET CONSOLE ON
93 :
94 : IF H=1 .AND. NUM = [4]
95 :     NUM = [5]
96 : ENDIF
97 :
98 : DO CASE
99 : CASE NUM='1'
100 :     SELE 5
101 :     * USE TEMP
102 :     APPEND BLANK
103 :     REPLACE PNUM WITH NPNUM
104 :     REPLACE DATE WITH NDATE
105 :     REPLACE ARTICLE WITH NARTICLE
106 :     REPLACE PARTNAME WITH NPARTNAME
107 :     REPLACE PLT WITH NPLT
108 :     REPLACE CODE WITH UPPER(NSHORT)
109 :     REPLACE ROUTCODE WITH UPPER(Z)
110 :     REPLACE ALTCODE WITH UPPER(ZZ)
111 :     IF NPTYP='SA'
112 :         REPLACE SACODE WITH UPPER(NSACODE)
113 :     ENDIF
114 :
115 :     RELEASE NUM,H,D,V,W,X,Y,Z,ZZ
116 :     RETURN
117 :
118 : CASE NUM='2'
119 :     LOOP
120 :
121 : CASE NUM='3'
122 :     CLEAR
123 :
124 :     SELE 6
125 :     * USE OPCODE LIST TO SEE DATA
126 :     DO L_OPCODE
```



B:MANUPLAN.PRG

Date: 25 January 1989 Time: 02:18:53 P-4

---

```
127 :
128 :     SELE 4
129 :     LOOP
130 :
131 :     CASE NUM='4'
132 :         H = 1
133 :         LOOP
134 :
135 :     CASE NUM='5'
136 :         CLEAR
137 :         CLEAR ALL
138 :         SET DELETED OFF
139 :         CANCEL
140 :
141 :     OTHERWISE
142 :         CLEAR
143 :
144 :         IF H=0
145 :             @ 12,25 SAY "YOU MUST ENTER A 1,2,3,4, OR 5"
146 :         ELSE
147 :             @ 12,25 SAY "YOU MUST ENTER A 1,2,3, OR 4"
148 :         ENDIF
149 :
150 :         @ 14,27 SAY "HIT ANY KEY TO CONTINUE"
151 :         SET CONSOLE OFF
152 :         WAIT
153 :         SET CONSOLE ON
154 :         LOOP
155 :     ENDCASE
156 : ENDDO
```

---

B:NEWADD.PRG

Date: 25 January 1989

Time: 02:09:49 P-1

```
-----
1 : CLEA ALL
2 : SET TALK OFF
3 : SET ECHO OFF
4 : SET BELL OFF
5 : SET DELETED ON
6 : SET EXACT ON
7 :
8 : CLEA
9 : @ 12,23 SAY [UPDATING STDPLAN.DBF - PLEASE WAIT]
10 :
11 : *----- OPEN FILE -----
12 : SELE 7
13 : USE ALTROUT
14 :
15 : SELE 2
16 : USE MAIN
17 :
18 : SELE 3
19 : USE MAINSA
20 :
21 : SELE 4
22 : USE STDPLAN
23 :
24 : SELE 5
25 : USE TEMP
26 : GO BOTTOM
27 : TOTAL_REC = RECNO()
28 : GO TOP
29 : *-----
30 :
31 : * SET UP A LOOP TO EXTRACT RECORDS FROM TEMP.DBF & UPDATE STDPLAN.DBF
32 :
33 : @ 8,28 SAY [TOTAL RECORDS      = ]+STR(TOTAL_REC,3)
34 : @ 10,28 SAY [WORKING ON RECORD = ]
35 : CC = COL()
36 : DO WHILE .NOT. EOF()
37 :   @ 10,CC SAY STR(RECNO(),3)
38 :   TPNUM      = PNUM
39 :   TCODE      = CODE
40 :   TROUTCODE  = ROUTCODE
41 :   TALTCODE   = ALTCODE
42 :   TSACODE    = SACODE
```

B:NEWADD.PRG

Date: 25 January 1989

Time: 02:10:01 P-2

```
-----
43 :
44 :   SELE 4
45 :   * USE STDPLAN
46 :   LOCATE FOR STDPLAN = TROUTCODE
47 :   REC = RECNO()
48 :   * IF NO MATCH IS FOUND FOR THE ROUTING SEQUENCE, THE ROUTCODE IS
49 :   * UNIQUE AND SHOULD BE GIVEN THE NEXT HIGHEST STDNUM AND ADDED
50 :   * TO THE FILE STDPLAN.DBF
51 :
52 :   IF EOF()
53 :     GO BOTTOM
54 :     NEWNUM = STR((VAL(STDPLAN->STDNUM)+1),5)
55 :     APPE BLANK
56 :   ELSE
57 :     * IF A MATCH IS FOUND FOR THE ROUTCODE, ASSIGN THE STDNUM TO
58 :     * THE NEW PART
59 :     GO REC
60 :     NEWNUM = STDPLAN->STDNUM
61 :     INSERT BLANK
62 :   ENDIF
63 :   REPLACE PNUM WITH TPNUM
64 :   REPLACE STDPLAN WITH TROUTCODE
65 :   REPLACE STDNUM WITH NEWNUM
66 :
67 :   SELE 5
68 :   REPL STDNUM WITH NEWNUM
69 :   SKIP
70 : ENDDO
71 :
72 : CLEA
73 : @ 6,29 SAY [TOTAL RECORDS      = ]+STR(TOTAL_REC,3)
74 : @ 8,29 SAY [WORKING ON RECORD = ]
75 : CC = COL()
76 : @ 10,26 SAY [STDPLAN.DBF HAS BEEN UPDATED]
77 : @ 12,26 SAY [UPDATING ALTROUT - PLEASE WAIT]
78 :
79 : SELE 5
80 : * SET UP A LOOP TO EXTRACT RECORDS FROM TEMP.DBF AND UPDATE ALTROUT.DBF
81 : * USE TEMP
82 : GO TOP
83 :
84 : DO WHILE .NOT. EOF()
```

B:NEWADD.PRG

Date: 25 January 1989

Time: 02:10:14 P-3

```
-----
85 : @ 8,CC SAY STR(RECNO(),3)
86 : TPNUM = PNUM
87 : TALTCODE = ALTCODE
88 :
89 : IF LEN(TRIM(TALTCODE)) = 0
90 :     SKIP
91 :     LOOP
92 : ENDIF
93 :
94 : SELE 7
95 : * USE ALTROUT
96 : APPEND BLANK
97 : REPLACE PNUM WITH TPNUM
98 : REPLACE ALTROUT WITH TALTCODE
99 :
100 : SELE 5
101 : SKIP
102 : ENDDO
103 :
104 : * SET UP A LOOP TO EXTRACT RECORDS FROM TEMP.DBF AND UPDATE
105 : * THE FILES MAIN.DBF AND MAINSA.DBF RECORDS IN TEMP.DBF WILL BE
106 : * DELETED AS THEY ARE PROCESSED. AT EOF, TEMP WILL BE PACKED,
107 : * MAIN.DBF WILL BE INDEXED ON PNUM(MAINPNUM.NDX) AND ON CODE
108 : * (MAINCODE.NDX), MAINSA.DBF WILL BE INDEXED ON PNUM (SAPNUM.NDX)
109 :
110 : CLEA
111 : @ 6,29 SAY [TOTAL RECORDS      = ]+STR(TOTAL_REC,3)
112 : @ 8,29 SAY [WORKING ON RECORD = ]
113 : CC = COL()
114 : @ 10,26 SAY [STDPLAN.DBF HAS BEEN UPDATED]
115 : @ 12,26 SAY [ALTROUT.DBF HAS BEEN UPDATED]
116 : @ 14,26 SAY [UPDATING MAIN.DBF - PLEASE WAIT]
117 :
118 : SELE 5
119 : * USE TEMP
120 : GO TOP
121 :
122 : DO WHILE .NOT. EOF()
123 :     @ 8,CC SAY STR(RECNO(),3)
124 :     TPNUM = PNUM
125 :     TDATE = DATE
126 :     TSTDNUM = STDNUM
```

B:NEWADD.PRG

Date: 25 January 1989

Time: 02:10:26 P-4

```
-----
127 :   TPLT = PLT
128 :   TCODE = CODE
129 :   TROUTCODE = ROUTCODE
130 :   TSACODE = SACODE
131 :   TARTICLE = ARTICLE
132 :   TPARTNAME = PARTNAME
133 :
134 :   * TEST TO SEE IF AN ALTERNATE ROUTING IS PRESENT. IF YES SET THE
135 :   * FIELD ALT TO [Y] IN MAIN.DBF, OTHERWISE SET ALT TO [N]
136 :
137 :   TALTCODE = ALTCODE
138 :
139 :   IF TALTCODE = [ ]
140 :       ALTFLAG = [N]
141 :   ELSE
142 :       ALTFLAG = [Y]
143 :   ENDIF
144 :
145 :   * BREAK DOWN THE FIELD [CODE] INTO ITS COMPONENT CODE SECTIONS
146 :
147 :   TRDIM = SUBSTR(TCODE,1,3)
148 :   TFDIM = SUBSTR(TCODE,4,3)
149 :   TPTYP = SUBSTR(TCODE,7,2)
150 :   THAKE = SUBSTR(TCODE,9,1)
151 :   THAT = SUBSTR(TCODE,10,5)
152 :   TGEO = SUBSTR(TCODE,15,25)
153 :
154 :   * ENTER THE NEWPART INFORMATION
155 :
156 :   SELE 2
157 :   * USE MAIN
158 :   APPEND BLANK
159 :
160 :   REPLACE PNUM WITH TPNUM
161 :   REPLACE STDNUM WITH TSTDNUM
162 :   REPLACE DATE WITH TDATE
163 :   REPLACE PLT WITH TPLT
164 :   REPLACE RDIM WITH TRDIM
165 :   REPLACE FDIM WITH TFDIM
166 :   REPLACE PTYP WITH TPTYP
167 :   REPLACE MAKE WITH THAKE
168 :   REPLACE MAT WITH THAT
```

B:NEWADD.PRG

Date: 25 January 1989

Time: 02:10:39 P-5

```
-----
169 | REPLACE GEO WITH TGEO
170 | REPLACE CODE WITH TCODE
171 | REPLACE ALT WITH ALTFLAG
172 | REPLACE ARTICLE WITH TARTICLE
173 | REPLACE PARTNAME WITH TPARTNAME
174 | REPLACE ROUTCODE WITH TROUTCODE
175 |
176 | * IF AN SACODE IS PRESENT, UPDATE MAINSA.DBF
177 |
178 | IF TSACODE <> [ ]
179 |     SELE 3
180 |     * USE MAINSA
181 |     APPEND BLANK
182 |
183 |     REPLACE PNUM WITH TPNUM
184 |     REPLACE PTYP WITH [SA]
185 |     REPLACE SACODE WITH TSACODE
186 | ENDIF
187 |
188 | SELE 5
189 | * USE TEMP
190 | SKIP
191 | ENDDO
192 |
193 | * ALL UPDATING IS COMPLETE, NOW EACH INDEX MUST BE UPDATED
194 |
195 | CLEA ALL
196 | CLEA
197 | @ 6,26 SAY [ALL FILES HAVE BEEN UPDATED]
198 | @ 8,26 SAY [RE-INDEXING IS NOW PROCEEDING]
199 |
200 | USE TEMP
201 | SET SAFE OFF
202 | PACK
203 | SET SAFE ON
204 | USE
205 | *-----
206 | SET SAFE OFF
207 | SET TALK ON
208 |
209 | @ 11,20 SAY [USE MAIN ----> INDEX ON CODE TO MAINCODE]
210 | USE MAIN
```

B:NEWADD.PRG

Date: 25 January 1989

Time: 02:10:51 P-6

---

211 : INDEX ON CODE TO MAINCODE  
212 : @ 12,0 CLEA  
213 : @ 12,20 SAY [ ---> INDEX ON CODE TO MAINPNUM ]  
214 : INDEX ON PNUM TO MAINPNUM  
215 : USE  
216 :  
217 : @ 13,0 CLEA  
218 : @ 14,20 SAY [USE MAINSA ---> INDEX ON PNUM TO SAPNUM]  
219 : USE MAINSA  
220 : INDEX ON PNUM TO SAPNUM  
221 : USE  
222 :  
223 : @ 15,0 CLEA  
224 : @ 16,20 SAY [USE ALTROUT ---> INDEX ON PNUM TO ALTPNUM]  
225 : USE ALTROUT  
226 : INDEX ON PNUM TO ALTPNUM  
227 : USE  
228 :  
229 : SET SAFE ON  
230 : \*-----  
231 :  
232 : RELEASE ALL  
233 : CLEA ALL  
234 : SET DELETED OFF  
235 : SET TALK ON  
236 : @ 23,0 SAY [PROGRAM TERMINATED...]  
237 : CANCEL

---

ภาคผนวก จ.

แบบฟอร์มวางแผนการผลิต



แบบฟอร์มวางแผนงานประจำวัน

ชื่อแผนก \_\_\_\_\_

วัน เดือน ปี \_\_\_\_\_

สายงาน	ชื่อแผนก											หมายเหตุ			
	8.00	9	10	11	12	13	14	15	16	17					
1	แผน														
	HR														
2	แผน														
	HR														
3	แผน														
	HR														

หัวหน้าแผนก \_\_\_\_\_

## แบบฟอร์มวางแผนงานประจำวันแผนกคัลหยาบ

วัน เดือน ปี \_\_\_\_\_

เครื่อง	เวลา	เช้า					บ่าย					หมายเหตุ
		8	9	10	11	12	1	2	3	4	5	
1	แผน											
	ผล											
2	แผน											
	ผล											
3	แผน											
	ผล											
4	แผน											
	ผล											
5	แผน											
	ผล											

หัวหน้าแผนก \_\_\_\_\_

แบบฟอร์ม ว่างแผนงานประจำวัน แผน ไส้กั่ว

วัน เดือน ปี \_\_\_\_\_

เวลา กิจกรรม	8	9	10	11	12	1	2	3	4	5
	เข้า									
1	แผน									
	ผล									
2	แผน									
	ผล									

หมายเหตุ \_\_\_\_\_

แบบฟอร์มวางแผนงานประจำวันแบบ คัดละเอียดยก

วัน เดือน ปี \_\_\_\_\_

เวลา เครื่อง	เช้า						บ่าย			หมายเหตุ	
	8	9	10	11	12	1	2	3	4		5
1	แทน										
	ทด										
2	แทน										
	ทด										
3	แทน										
	ทด										

หัวหน้าแผนก \_\_\_\_\_

## แบบฟอร์มวางแผนงานประจำวัน แผนกปอก / เจาะเคื่อย

วัน เดือน ปี \_\_\_\_\_

เวลา เครื่อง	เช้า					12	บ่าย				
	8	9	10	11	12		1	2	3	4	5
ปอกเคื่อย รูปไข่ 1	แผน										
	ผล										
ปอกเคื่อย รูปไข่ 2	แผน										
	ผล										
ปอกเคื่อย 2 หัว	แผน										
	ผล										

หัวหน้าแผนก \_\_\_\_\_

แบบฟอร์มวางแผนงานประจำวันแบบ ปอก/เจาะเคียว

วัน เดือน ปี \_\_\_\_\_

เวลา เครื่อง	เช้า						พัก	บ่าย				
	8	9	10	11	12	1		2	3	4	5	
เจาะเคียว รูปไข่ 1	แทน											
	ผล											
เจาะ เคียว รูปไข่ 2	แทน											
	ผล											
เจาะ เคียว รูปไข่ แบบหัว	แทน											
	ผล											
เครื่อง เจาะรู แนวตั้ง หลายหัว 1	แทน											
	ผล											
เครื่อง เจาะรู แนวตั้ง หลายหัว 2	แทน											
	ผล											
เครื่อง เจาะเคียว กลมนอน 2 หัว	แทน											
	ผล											

หัวหน้าแผนก \_\_\_\_\_

แบบฟอร์มวางแผนงานประจำวัน แผนกเลี้ยงวัว

วัน เดือน ปี \_\_\_\_\_

เวลา เครื่อง	เช้า					12	บ่าย				
	8	9	10	11	12		1	2	3	4	5
3	แผน					12					
	ผล										
4	แผน										
	ผล										
5	แผน										
	ผล										
6	แผน										
	ผล										

หัวหน้าแผนก \_\_\_\_\_

แบบฟอร์มวางแผนงานประจำวันแผนก เหล้าหึ่ง

วัน เดือน ปี \_\_\_\_\_

เวลา เครื่อง	เช้า					บ่าย					กลางคืน				
	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10
1	แผน														
	ผล														
2	แผน														
	ผล														
3	แผน														
	ผล														
4	แผน														
	ผล														
5	แผน														
	ผล														
6	แผน														
	ผล														
7	แผน														
	ผล														
8	แผน														
	ผล														

หัวหน้าแผนก \_\_\_\_\_



แบบฟอร์มวางแผนงานประจำวัน แผนกกลิ้ง

วัน เดือน ปี \_\_\_\_\_

เวลา เครื่อง	เช้า					12	บ่าย					
	8	9	10	11	12		1	2	3	4	5	
กึ่งแป้น												
เลส												
กึ่งแป้น												
ชอปเปอร์												
เครื่อง												
กลิ้ง												

หัวหน้าแผนก \_\_\_\_\_



แผ่น 2

เวลา เครื่อง	เช้า					12	บ่าย					กลางคืน			
	8	9	10	11	12		1	2	3	4	5	6	7	8	9
11															
12															
13															
14															
15															





แบบฟอร์มวางแผนงานประจำวัน

ชื่อแผนก \_\_\_\_\_ วัน เดือน ปี \_\_\_\_\_

สายงาน	วัน เดือน ปี											หมายเหตุ		
	8.00	9	10	11	12	13	14	15	16	17				
1	แผน													
	สด													
2	แผน													
	สด													
3	แผน													
	สด													

ทำหน้าแทน



188

ประวัติผู้เขียน

ข้าพเจ้า นายสมชาย สงวนศักดิ์ เกิดวันที่ 15 กันยายน 2502 ที่จังหวัดชลบุรี  
จบการศึกษาระดับปริญญาตรี สาขาวิชาเทคโนโลยีและอาชีวศึกษา วิทยาเขตเทเวศร์ ในปี  
พ.ศ. 2526 ปัจจุบันทำงานอยู่ที่สำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม กระทรวงอุตสาหกรรม