



HAL
open science

Un périphérique d'ordinateur à l'usage des musiciens : problèmes d'interprétation de la musique et d'édition de partitions

Denis Jaeger

► To cite this version:

Denis Jaeger. Un périphérique d'ordinateur à l'usage des musiciens : problèmes d'interprétation de la musique et d'édition de partitions. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1974. Français. NNT : . tel-00284720

HAL Id: tel-00284720

<https://theses.hal.science/tel-00284720>

Submitted on 3 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

pour obtenir le grade de
Docteur de Spécialité (3^e cycle)

Spécialité : INFORMATIQUE

par

Denis JAEGER

UN PERIPHERIQUE D'ORDINATEUR A L'USAGE DES MUSICIENS :
PROBLEMES D'INTERPRETATION DE LA MUSIQUE ET D'EDITION
DE PARTITIONS.

Thèse soutenue le 17 avril 1974 devant la commission d'examen :

Président : Monsieur J. KUNTZMANN

Examineurs : Monsieur B. Van CUTSEM
Monsieur A. LODÉON
Monsieur J. MERMET

Président : Monsieur Michel SOUTIF
Vice-Président : Monsieur Gabriel CAU

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Georges	Clinique des maladies infectieuses
	ARNAUD Paul	Chimie
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BENOIT Jean	Radioélectricité
	BERNARD Alain	Mathématiques Pures
	BESSON Jean	Electrochimie
	BEZES Henri	Chirurgie générale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BONNIER Etienne	Electrochimie Electrometallurgie
	BOUCHERLE André	Chimie et Toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques Appliquées
	BRAVARD Yves	Géographie
	BRISSONNEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	Electronique
	CABANAC Jean	Pathologie chirurgicale
	CABANEL Jean	Clinique rhumatologique et hydrologie
	CALAS François	Anatomie
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et Toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Oto-Rhino-Laryngologie
	CHATEAU Robert	Thérapeutique
	CHENE Marcel	Chimie papetière
	COEUR André	Pharmacie chimique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie Pathologique
	CRAYA Antoine	Mécanique

Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DESRE Pierre	Métallurgie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée
	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de Dermatologie et Syphiligraphie
	FAU René	Clinique neuro-psychiatrique
	FELICI Noël	Electrostatique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques Pures
	GALVANI Octave	Mathématiques Pures
	GASTINEL Noël	Analyse numérique
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques Pures
	GIRAUD Pierre	Géologie
	KLEIN Joseph	Mathématiques Pures
Mme	KOFLER Lucie	Botanique et Physiologie végétale
MM.	KOSZUL Jean-Louis	Mathématiques Pures
	KRAVTCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre-Jean	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LLIBOUTRY Louis	Géophysique
	LOUP Jean	Géographie
Mlle	LUTZ Elisabeth	Mathématiques Pures
MM.	MALGRANGE Bernard	Mathématiques Pures
	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MASSEPORT Jean	Géographie
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et Pétrographie
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	NEEL Louis	Physique du solide
	OZENDA Paul	Botanique
	PAUTHENET René	Electrotechnique
	PAYAN Jean-Jacques	Mathématiques Pures
	PEBAY-PEYROULA Jean-Claude	Physique
	PERRET René	Servomécanismes
	PILLET Emile	Physique industrielle
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	REULOS René	Physique industrielle
	RINALDI Renaud	Physique
	ROGET Jean	Clinique de pédiatrie et de puériculture
	SANTON Lucien	Mécanique
	SEIGNEURIN Raymond	Microbiologie et Hygiène
	SENGEL Philippe	Zoologie
	SILBERT Robert	Mécanique des fluides
	SOUTIF Michel	Physique générale

MM.	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLAND François	Zoologie
	VALENTIN Jacques	Physique nucléaire
	VAUQUOIS Bernard	Calcul électronique
Mme	VERAIN Alice	Pharmacie galénique
M.	VERAIN André	Physique
Mme	VEYRET Germaine	Géographie
MM.	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM.	BULLEMER Bernhard	Physique
	HANO JUN-ICHI	Mathématiques Pures
	STEPHENS Michaël	Mathématiques appliquées

PROFESSEURS SANS CHAIRE

MM.	BEAUDOING André	Pédiatrie
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques appliquées
	BIAREZ Jean-Pierre	Mécanique
	BONNETAIN Lucien	Chimie minérale
Mme	BONNIER Jane	Chimie générale
MM.	CARLIER Georges	Biologie végétale
	COHEN Joseph	Electrotechnique
	COUMES André	Radioélectricité
	DEPASSEL Roger	Mécanique des fluides
	DEPORTES Charles	Chimie minérale
	GAUTHIER Yves	Sciences biologiques
	GAVEND Michel	Pharmacologie
	GERMAIN Jean-Pierre	Mécanique
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	HACQUES Gérard	Calcul numérique
	JANIN Bernard	Géographie
Mme	KAHANE Josette	Physique
MM.	MULLER Jean-Michel	Thérapeutique
	PERRIAUX Jean-Jacques	Géologie et Minéralogie
	POULOUJADOFF Michel	Electrotechnique
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	ROBERT André	Chimie papetière
	DE ROUGEMONT Jacques	Neurochirurgie
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIBILLE Robert	Construction mécanique
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

Mle	AGNIUS-DELORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBLARD Pierre	Dermatologie
	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Yves	Chimie
	BEGUIN Claude	Chimie organique
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
	BILLET Jean	Géographie
	BLIMAN Samuel	Electronique (EIE)
	BLOCH Daniel	Electrotechnique
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BOUCHET Yves	Anatomie
	BOUVARD Maurice	Mécanique des fluides
	BRODEAU François	Mathématiques (IUT B)
	BRUGEL Lucien	Energétique
	BUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHIAVERINA Jean	Biologie appliquée (EFP)
	CHIBON Pierre	Biologie animale
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CONTE René	Physique
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	DURAND Francis	Métallurgie
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	GENSAC Pierre	Botanique
	GIDON Maurice	Géologie
	GRIFFITHS Michaël	Mathématiques appliquées
	GROULADE Joseph	Biochimie médicale
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et Médecine préventive
	IDELMAN Simon	Physiologie animale
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	JOLY Jean-René	Mathématiques Pures
	JOUBERT Jean-Claude	Physique du solide
	JULLIEN Pierre	Mathématiques Pures
	KAHANE André	Physique générale
	KUHN Gérard	Physique
	LACOUME Jean-Louis	Physique
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LANCIA Roland	Physique atomique
	LE JUNTER Noël	Electronique
	LEROY Philippe	Mathématiques
	LOISEAUX Jean-Marie	Physique nucléaire
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LUU DUC Cuong	Chimie organique
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et Médecine préventive
	MARECHAL Jean	Mécanique
	MARTIN-BOUYER Michel	Chimie (CUS)

MM.	MAYNARD Roger	Physique du solide
	MICHOULIER Jean	Physique (IUT A)
	MICOUD Max	Maladies infectieuses
	MOREAU René	Hydraulique (INP)
	NEGRE Robert	Mécanique
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B)
	PEFFEN René	Métallurgie
	PELMONT Jean	Physiologie animale
	PERRET Jean	Neurologie
	PERRIN Louis	Pathologie expérimentale
	PFISTER Jean-Claude	Physique du solide
	PHELIP Xavier	Rhumatologie
Mlle	RIERY Yvette	Biologie animale
MM.	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RENAUD Maurice	Chimie
	RICHARD Lucien	Botanique
Mme	RINAUDO Marquerite	Chimie macromoléculaire
MM.	ROMIER Guy	Mathématiques (IUT B)
	SHOM Jean-Claude	Chimie générale
	STIEGLITZ Paul	Anesthésiologie
	STOEBNER Pierre	Anatomie pathologique
	VAN CUTSEM Bernard	Mathématiques appliquées
	VEILLON Gérard	Mathématiques appliquées (INP)
	VIALON Pierre	Géologie
	VOOG Robert	Médecine interne
	VROUSSOS Constantin	Radiologie
	ZADWORNY François	Electronique

MAITRES DE CONFERENCES ASSOCIES

MM.	BOUDOURIS Georges	Radioélectricité
	CHEEKE John	Thermodynamique
	GOLDSCHMIDT Hubert	Mathématiques
	SIDNEY STUARD	Mathématiques Pures
	YACOUD Mahmoud	Médecine légale

CHARGES DE FONCTIONS DE MAITRES DE CONFERENCES

Mme	BERIEL Hélène	Physiologie
Mme	RENAUDET Jacqueline	Microbiologie

Fait le 30 mai 1972.

Le travail qui a fait l'objet de ce mémoire a été réalisé dans le laboratoire de l'E.N.S.I.M.A.G. dirigé par Monsieur le Professeur Kuntzmann qui m'a fait l'honneur de présider le jury de ma thèse, je le prie de recevoir ici l'expression de ma profonde reconnaissance.

A Monsieur le Professeur Van Cutsem qui s'est intéressé à mon travail et a bien voulu accepter de faire partie du jury, je tiens à exprimer ici ma respectueuse gratitude.

Je tiens aussi à remercier particulièrement Monsieur Lodéon, Directeur de l'Ecole Nationale de Musique et d'Art Dramatique de Grenoble, pour l'intérêt qu'il porte à mes recherches et l'honneur qu'il m'a fait en acceptant de faire partie du jury auxquels j'ai été très sensible.

A Monsieur Mermet, qui m'a guidé dans mes recherches et en a suivi pas à pas le développement, j'exprime mes remerciements les plus vifs pour les nombreux conseils qu'il m'a prodigués.

Toutes les personnes de l'E.N.S.I.M.A.G., notamment Messieurs Fantino et Saillard, ou de l'agence I.B.M. de Grenoble qui, par leurs conseils ou leur aide, ont contribué à la réalisation de ce mémoire, trouveront ici l'expression de ma sincère reconnaissance.

A Mademoiselle Denarier qui a assuré la frappe du manuscrit, au secrétariat et au service de tirage, j'adresse mes remerciements très cordiaux.

CHAPITRE I : INTRODUCTION

I - § 1) GENERALITES

L'idée de départ de la présente recherche est de mettre à la disposition des musiciens (conservatoires, maisons d'édition, etc...) un dispositif d'édition automatique de la musique pour pallier aux problèmes laborieux actuels de la transcription manuelle. L'originalité du système réside dans le fait qu'il est proposé non à des informaticiens, mais à des musiciens qui peuvent être totalement ignorants des problèmes informatiques. En effet, ce dispositif sera composé grossièrement d'un ordinateur auquel seront reliés deux organes d'entrée (un clavier de piano et un clavier alphanumérique pour contrôle) et un organe de sortie pour la partition éditée (un écran de télévision ou une imprimante). Un air de musique joué sur le clavier de piano devra, après traitement par le ordinateur, être édité sous forme de partition musicale sur l'organe de sortie.

Il est reconnu que la Musique et les Mathématiques ont de nombreux points communs [7] [8] et, dès l'apparition des ordinateurs, beaucoup de gens se sont intéressés à voir ce que pouvait apporter l'Informatique dans le domaine de la recherche musicale ; ce domaine peut se diviser en plusieurs secteurs : la composition musicale [6] , l'analyse de style, la génération de son [2] , la lecture de la musique et l'édition de la musique. La lecture et l'édition de la musique sont sans aucun doute, les secteurs les moins explorés, car la musique est alors considérée comme langage ayant une certaine syntaxe, abstraction faite de toute signification musicale. Beaucoup de langages (tous linéaires) de représentation de la musique en mémoire ont été définis [11] ; notons au passage que, par opposition à l'écriture musicale habituelle qui est à deux dimensions, nous appelons langage linéaire un langage dont les phrases sont constituées de symboles rangés les uns à la suite des autres suivant une direction donnée ; les partitions à analyser devaient, en tout cas à l'origine, être codées à la main, dans un de ces langages, avant traitement par le ordinateur et le résultat de compositions musicales

(à l'aide d'un ordinateur) devait être décodé à la main. A notre connaissance, quelques projets d'édition de la musique ont été réalisés sur table traçante ou par procédé de photocomposition ; les projets prévoyant l'utilisation d'une imprimante semblent délaissés ou progressent lentement [9] [10]. Quant à la lecture de la musique, le seul projet à notre connaissance est un projet de lecture directe de partitions par dispositif optique [11].

1 - § 2) MODE DE FONCTIONNEMENT DU SYSTEME PROPOSE

Si la saisie des données relatives à la hauteur des notes semble à priori ne poser aucun problème, il n'en est pas de même pour la saisie de la durée des notes, car aucun interprète ne peut atteindre la précision d'une mécanique et on est donc amené à faire certaines approximations. De ce fait, si nous voulons traduire les durées que l'interprète a effectivement voulu exprimer et non les durées réelles qu'un dispositif d'horloge nous permet de mesurer, nous nous trouvons souvent face à des ambiguïtés. Ceci nous a amené à envisager deux modes de fonctionnement.

1) Dans le premier cas on peut prévoir de rajouter un certain nombre de pédales au clavier de piano. Ces pédales permettront d'indiquer, pour chaque note ou accord joué, la durée (ronde, blanche, ...). Il va sans dire que, par exemple, une série de croches successives pourra être jouée en maintenant enfoncée la pédale "croche". Il faut aussi, dans ce cas, un certain nombre de touches pour indiquer les silences, les caractères spéciaux, ... Ce premier mode de fonctionnement simplifie beaucoup l'interprétation par le calculateur et permet de faire éditer une partition par le calculateur sous la forme souhaitée. Ce mode intéressera donc surtout les éditeurs et les musicologues.

2) Pour le second mode on peut prévoir un métronome électrique pour imposer à l'interprète de respecter à peu près un certain rythme. Dans ce mode, l'interprète jouera donc un morceau sans s'arrêter, l'interprétation de la durée des notes étant faite par le calculateur. On peut prévoir des programmes d'approximation des durées ; plus ces programmes seront sophistiqués, plus on se rapprochera d'une édition idéale répondant aux désirs de l'interprète-éditeur et dans ce cas le premier mode de fonctionnement pourrait être supprimé ; moins ces programmes seront sophistiqués et plus la partition résultante sera fidèle au jeu de l'interprète. Dans ce mode de

fonctionnement, la fidélité au jeu de l'interprète prime sur la qualité de la partition résultante ; ceci intéressera donc en priorité les compositeurs - pianistes (ou organistes).



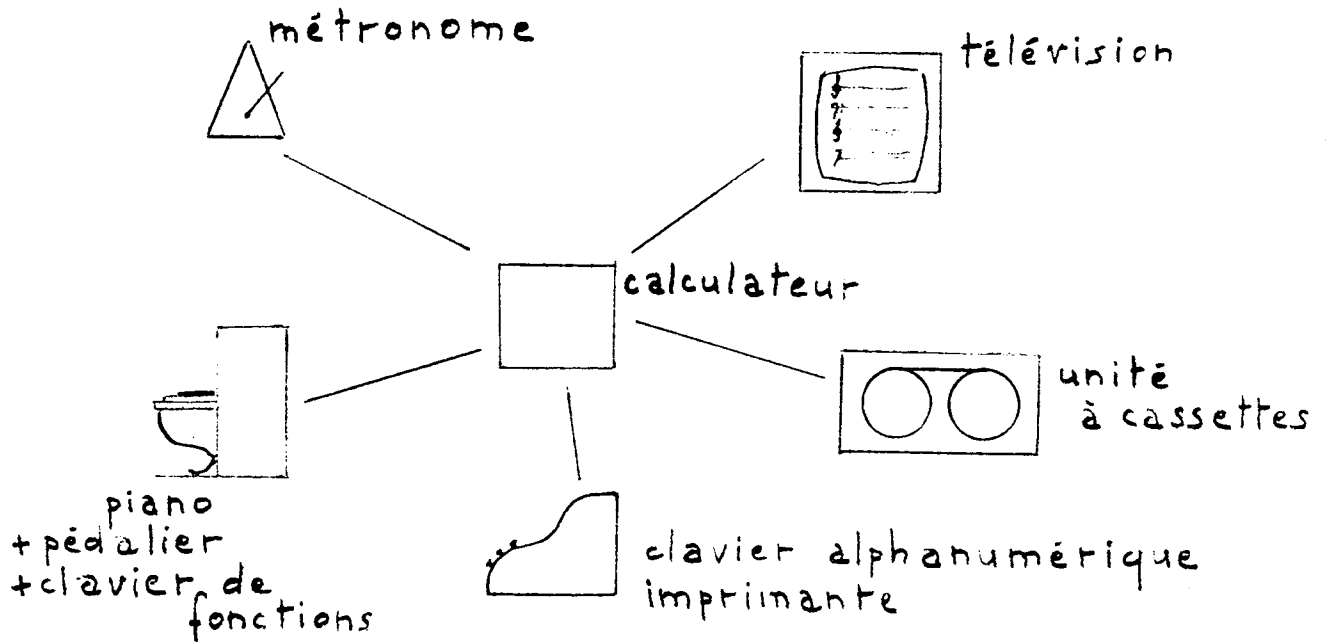
CHAPITRE II : DESCRIPTION GENERALE DU PERIPHERIQUE

Pour permettre de mieux comprendre le fonctionnement du système, nous citerons ici les dispositifs technologiques que devra comprendre le prototype, sans donner toutefois beaucoup d'indications sur le hardware ; pour plus de détails à ce sujet, se reporter à l'annexe I où un développement plus substantiel y est fait.

II - § 1) CONFIGURATION GENERALE

Il ne faut pas perdre de vue que le système proposé n'est pas un instrument de recherche pure, mais un instrument d'édition automatique destiné à rendre service aux musicologues et musiciens (compositeurs et éditeurs). Il est donc nécessaire d'étudier une configuration suffisamment rentable pour être séduisante. Un terminal musical, relié à un gros ordinateur travaillant en temps partagé, est une solution qui ne semble pas réunir beaucoup de suffrages, malgré l'attrait d'un espace - mémoire important, pourtant souhaitable du fait de l'importance des programmes musicaux ; car le volume des transferts d'informations entre le terminal et le calculateur est trop grand pour espérer un temps de réponse acceptable. Nous avons préféré, quitte à sectionner ou mettre en mémoire morte une partie des programmes, proposer un système autonome comprenant un petit ordinateur spécialisé. Cela ne doit pas exclure une éventuelle connexion de l'ensemble à un gros ordinateur ou mieux, la possibilité d'exécuter, à l'aide d'un gros ordinateur non connecté à l'ensemble, certains traitements sur les partitions créées. Nous conserverons, pour le système, le nom de périphérique.

II - 1 - a) DESCRIPTION DE LA CONFIGURATION



- un calculateur (4 - 20 K - octets)
- un clavier de piano (61 touches)
- un clavier de fonctions (57 touches)
- un pédalier (10 pédales)
- un métronome électrique
- un écran de télévision
- un clavier alphanumérique
- une imprimante
- un lecteur - enregistreur de cassettes
-

II - 1 - b) CLAVIER DE PIANO

Des contacts placés sous les touches du clavier de piano permettent de signaler au calculateur les hauteurs et durées (temps écoulé entre l'enfoncement et le relachement des touches) des notes.

II - 1 - c) CLAVIER DE FONCTIONS

Pour permettre l'exécution de fonctions particulières et l'édition d'informations musicales spéciales (caractères spéciaux, commentaires) un clavier de fonctions sera utilisé (voir annexe IV).

II - 1 - d) PEDALIER

Si la qualité des résultats des programmes d'approximation de la durée des notes ne satisfait pas l'utilisateur (en particulier le musicien-éditeur), un second moyen d'entrée des durées lui est proposé : un jeu de 7 pédales pour indiquer "ronde", "blanche", "noire", ..., "quadruple croche". Trois autres pédales ont été définies et correspondent aux fonctions de L.I. : "REL", "ACC" et "ACA" (voir chap. IV - 2 - b).

II - 1 - e) METRONOME ELECTRIQUE

Pour obtenir le meilleur rendement des programmes d'approximation des durées des notes, il sera possible de synchroniser l'interprète et le calculateur au moyen d'un métronome électrique.

II - 1 - f) ECRAN DE TELEVISION

Pour transmettre en temps réel (ou légèrement différé), après traitement par les programmes d'édition, les partitions créées par l'utilisateur.

II - 1 - g) CLAVIER ALPHANUMERIQUE

Le clavier alphanumérique sera utilisé pour définir les différents travaux demandés (création de partition, mise à jour, ...) et préciser certaines fonctions du clavier de fonctions.

II - 1 - h) IMPRIMANTE

Pour l'impression des messages d'erreurs et de contrôle, et l'impression des partitions en temps différé.

II - 1 - i) AUTRES DISPOSITIFS

La liste des dispositifs n'est pas limitative, on peut en particulier prévoir des unités pouvant servir de bibliothèques pour les partitions (enregistreur à cassettes par exemple).

II - § 2) PRINCIPES GENERAUX DE FONCTIONNEMENT

4 modes de fonctionnement sont à la disposition de l'utilisateur (voir chap.III):

- un mode pour la direction des travaux et l'exécution de certains programmes (de traitement des partitions) ne nécessitant pas d'intervention manuelle,

- deux modes de création des partitions ; ces modes correspondent aux deux modes annoncés dans le chap. I - 2 ,

- un mode de mise à jour des partitions.

Deux langages ont été définis pour représenter en mémoire les partitions créées :

- le langage interne (L.I.) dont les chaînes décrivent plus particulièrement les concepts musicaux des partitions,

- le langage de sortie (L.S.) dont les chaînes décrivent le dessin des partitions telles qu'elles doivent être éditées.

Une partition créée est traduite en L.I., puis en L.S. et enfin éditée.

La télévision et l'imprimante sont utilisées pour assurer l'édition respectivement en temps réel (au fur et à mesure de la création d'une partition) et en temps différé.

CHAPITRE III : LE LANGAGE DE COMMANDES

III - § 1) PRINCIPES GENERAUX

Le clavier alphanumérique et l'imprimante vont servir en premier lieu comme organes de direction et contrôle du déroulement des travaux. En fait quel que soit le type d'imprimante finalement choisie pour l'impression des partitions, il faudra lui associer le clavier alphanumérique, l'ensemble pouvant alors tenir lieu de console d'opérateur.

Quatre modes de travail seront à la disposition de l'utilisateur. Le mode, sous lequel on entre automatiquement après l'initialisation du système, est le mode 1 ; les ordres que l'on peut donner sous ce mode sont des ordres d'entrée sous les modes 2, 3 ou 4 ou des demandes d'exécution de fonctions spéciales servant à gérer les fichiers ou à leur appliquer des traitements pendant lesquels aucune intervention manuelle n'est requise. Après l'achèvement d'une fonction spéciale demandée ou après l'exécution d'un ordre de "sortie de mode" spécifique aux modes 2, 3 et 4, le contrôle est redonné au mode 1. En outre, il sera possible, à l'aide d'une touche de fonction ou d'une interruption manuelle (style ATTN des IBM 2741), de quitter à tout moment un mode ou d'interrompre l'exécution d'une fonction spéciale pour redonner le contrôle au mode 1. Cela suppose l'exécution, avant la reprise de contrôle par le mode 1, d'un certain nombre de procédures d'achèvement en catastrophe du travail interrompu (clôre les fichiers utilisés, par exemple). Les modes 2 et 3 correspondent respectivement aux premier et second modes annoncés dans le chapitre I. Le mode 4 est réservé à la mise à jour des partitions.

III - § 2) NOTION DE FICHIERS

Les partitions pourront être conservées en mémoire ou sur support extérieur (suivant la taille de la mémoire du calculateur) sous forme de fichiers. Ces fichiers auront un certain nombre d'enregistrements initiaux contenant leurs caractéristiques et seront de deux sortes : les "fichiers L.I." et les "fichiers L.S." Dans ces fichiers les partitions seront respectivement codées en langage L.I. (voir chap. IV) et en langage L.S. (voir chap. VI). Le langage L.I. est un langage décrivant des concepts musicaux et dont la traduction en DARMS peut être envisagée (cf. introduction du chap. IV et chap. IV - 4). Le langage L.S. est un langage plus proche de l'édition et dont les chaînes sont des suites de symboles (destinés à être imprimés) dont l'assemblage est une partition musicale ; aucun concept musical n'est décrit au moyen de ce langage et la traduction de ce langage en DARMS est, sinon impensable, du moins non réalisable dans un avenir proche. D'autre part, toute partition codée en L.I. doit, avant d'être éditée, être traduite en L.S. Il s'ensuit que des partitions pouvant encourir des modifications mettant en jeu les concepts musicaux (supprimer ou rajouter une note, changer de ton, etc...) ou susceptibles d'être traduites en DARMS en vue d'une analyse sémantique, doivent être conservées écrites en L.I. Par contre, les partitions définitives, destinées seulement à être éditées seront de préférence conservées écrites en L.S. pour éviter, à chaque édition, la traduction de leur forme L.I. en L.S.

III - 2 - a) FICHIERS L.I.

Les fichiers L.I. ont sept enregistrements initiaux contenant leurs caractéristiques :

- enregistrement titre
- enregistrement sous-titre
- enregistrement auteur
- enregistrement commentaire
- nombre d'enregistrements du fichier
- nombre de voix de la partition
- numéro de la voix

Les quatre premiers enregistrements sont des chaînes de caractères dont la longueur maximale est à définir lors de l'implémentation du système prototype,

mais ne doit toutefois pas excéder 100 caractères (nombre maximum prévu pour une ligne - télévision ou une ligne - imprimante). Ces quatre enregistrements devront contenir (sans que cela soit une obligation) le titre, le sous-titre, le nom de l'auteur de l'oeuvre et un commentaire, et serviront, à l'édition, d'en-tête pour la partition (le cadrage correct étant automatique).

Ex.

VALSE

"Valse de l'Adieu"

FR. CHOPIN Op. 69 - n° 1

Oeuvre posthume composée en 1836

Pour toute partition, il sera créé un nombre de fichiers L.I. égal au nombre de voix de la partition, chacun contenant la description d'une voix en L.I.

III - 2 - b) FICHIERS L.S.

Aucun concept musical n'étant décrit en L.S. la distinction entre les voix d'une partition n'est pas nécessaire ; il n'y aura donc toujours qu'un seul fichier L.S. pour une partition donnée. Par suite ces fichiers n'auront que cinq enregistrements initiaux contenant les mêmes renseignements que ceux des fichiers L.I. correspondants (N.B. : les quatre premiers seront strictement identiques).

III - § 3) LES MODES

III - 3 - a) MODE 1

III - 3 - a - 1 - Généralités

Pour pouvoir disposer d'un système suffisamment souple, un véritable langage de commandes doit être instauré et c'est sous le mode 1 que ces commandes seront invoquées. La plupart n'ont pas été étudiées en détail puisque ne posant pas ou peu de problèmes liés à la musique, mais des problèmes d'informatique souvent classiques. Nous en donnerons cependant un aperçu, sans nous y attarder, et verrons plus précisément les commandes d'impression et de création de fichiers.

On peut convenir que les deux enregistrements initiaux des fichiers L.I. et L.S. sont les deux caractéristiques essentielles des fichiers auxquelles feront référence la plupart des commandes.

Il faut prévoir des commandes d'effacement de fichiers, de concaténation de fichiers, de fin de session, ... qui sont des commandes utilitaires classiques.

Outre ces commandes indispensables, pourront être prévues des commandes d'exécution de fonctions opérant sur la géométrie des partitions :

. sans changer les effets musicaux :

- transposition dans un autre ton
- regroupement de voix
- inversion de voix
- changement de clés
- changement de mesures quand cela est possible (de 4/4 on peut passer à 4/2 : il suffit de doubler les durées des notes, mais de 3/4 il est difficile de passer à 4/4 sans changer l'effet musical)
- retassement (c'est-à-dire supprimer les espaces horizontaux en trop sans nuire à l'alignement des voix entre elles)
- cadrage :

Ex.



N.B. : il ne suffit pas de rajouter des espaces avant la dernière barre de mesure mais uniformément tout au long des portées et sans nuire à l'alignement des voix.

. en changeant les effets musicaux :

- changement des durées des notes sans tenir compte de la mesure :
ronde —→ quadruple croche
blanche —→ triple croche
- opérations de substitutions sur les valeurs des notes
- ...

N.B. ces dernières fonctions n'ont aucun rapport avec l'édition proprement dite.

III - 3 - a - 2 - Commande PRINT

PRINT titre, sous-titre, (option 1, ..., option n)

Les options permettent d'indiquer le lieu d'origine et le lieu de transfert de la partition ainsi que le type (L.I. ou L.S.) de la partition et dépendront de la configuration du système.

Ex.

type de transfert		travail exécuté
{ mémoire unité de stockage unité de stockage	→ unité de stockage	le fichier L.S. ou les fichiers L.I. (suivant le type précisé) de la partition sont transférés.
	→ mémoire	

{ mémoire unité de stockage	→ imprimante	si un fichier L.S. existe pour la partition celui-ci est imprimé sinon les fichiers L.I. sont traduits en L.S. puis imprimés

III - 3 - a - 3 - Commande EDITION

EDITION (titre, sous-titre { , auteur } , commentaire { }), m { , clé, ton }

Les paramètres entre accolades sont facultatifs. m, clé, ton indiquent respectivement le numéro de la voix, les clé et ton de la voix. Les paramètres titre,

sous-titre, auteur et commentaire sont des chaînes de caractères.

Si p fichiers L.I. (p voix), de même titre et sous-titre, existent déjà et $m \leq p$, alors la valeur que prend le pointeur horizontal (voir chap. IV - 2) au bout de la chaîne L.I. de la voix précisée est calculée et affectée au pointeur, la valeur du pointeur vertical est mise à 0. Clé et ton sont ignorés.

Si p fichiers existent ($p \geq 0$) et $m > p$, alors $m - p$ fichiers sont initialisés et les enregistrements "nombre de voix de la partition" prennent la valeur p . Les pointeurs horizontal et vertical prennent la valeur 0.

Le contrôle est alors passé au mode 2.

Au cours du rodage du système prototype et suivant les critiques des musiciens, il faudra prévoir de rajouter à la commande EDITION un certain nombre d'options, en particulier pour pouvoir indiquer si les valeurs des PV pour les fonctions soulignées du chap. IV doivent être prises par défaut ou chaque fois précisées par l'utilisateur (voir chap. IV et en particulier chap. IV - 2 - f)

III - 3 - a - 4 - Commande PIANO

PIANO (titre, soustitre { , auteur { , commentaire } }), clé, ton, mes

Commande utilisée pour entrer (après initialisation) sous le mode 3.

Pour cette commande, un seul fichier L.I. (une seule voix) est créé et aucun fichier L.I. de mêmes titre et sous-titre ne doit exister.

Les paramètres clé, ton et mes indiquent respectivement les clé, ton et mesure initiaux.

Le rythme est donné par le métronome électrique qui bat l'unité de temps.

N.B. : la seule unité d'entrée qui soit active sous le mode 1 est le clavier alphanumérique (exception faite des unités de stockage éventuellement utilisées par la commande PRINT).

III - 3 - b) MODE 2

Sont actives toutes les unités d'entrée sauf le métronome (clavier de piano + pédalier, clavier de fonctions et clavier alphanumérique).

Le clavier de piano est utilisé pour indiquer les notes jouées, le pédalier pour indiquer les durées des notes et les accords, le clavier de fonctions et le clavier alphanumérique pour indiquer les silences et symboles spéciaux (clés, altérations différentes de celles prises par défaut, doigtés, staccatos, points d'orgue, n-lets, etc...). De plus amples informations sont données dans le chapitre V.

L'édition est assurée en temps réel (ou légèrement différé sur l'écran de télévision).

La pression sur la touche ATTN du clavier alphanumérique provoque le retour au mode 1 après la suppression de la fonction L.I. en cours si elle est inachevée, la complétion de la traduction en L.S. de la chaîne L.I. créée et la clôture de tous les fichiers.

III - 3 - c) MODE 3

Les unités d'entrée seules actives sont le clavier de piano et le métronome électrique. La mise en marche du métronome provoque l'activation du clavier de piano. Le premier temps de la première mesure de la partition à créer débute sur le premier top de métronome suivant l'appui et le relâchement successifs d'une touche du clavier.

La pression sur la touche ATTN du clavier alphanumérique provoque la complétion de la mesure en cours au moyen de silences, la complétion de la traduction en L.S. de la chaîne L.I. créée et la clôture de tous les fichiers.

L'édition est assurée sur l'écran de télévision en temps réel au début et de plus en plus différé, car la taille de la file des informations d'entrée, en attente d'interprétation par les programmes de traduction et d'édition, ira en augmentant vues la rapidité et la complexité du flot d'entrée : un appui ou un relâchement d'une touche du clavier de piano enverra au calculateur les informations "n° du signal"

et "valeur de l'horloge" qui seront placées dans une file d'entrée dans laquelle sera également placée l'heure à chaque top de métronome.

ex.

Une partition contenant 15 notes à la seconde (y compris les notes des accords) est déjà une partition assez dense. Si nous avons en moyenne 2 tops de métronome par seconde, le nombre des informations en provenance du métronome sera négligeable devant le nombre des informations en provenance du clavier. Pour une note jouée nous aurons deux signaux d'entrée (appui et relâchement). Donc 15 notes par seconde impliquent 30 signaux d'entrée par seconde. Sachant que 3 octets sont largement suffisants pour contenir les informations "n° du signal" et "valeur de l'horloge", en une seconde nous aurons généré 90 octets, en une minute nous en aurons $90 \times 60 = 5400$, soit 5,4 K-octets ce qui n'est pas excessif compte tenu du fait qu'une partie de la file aura déjà été interprétée et que 15 notes à la seconde représentent, pour une partition, une densité supérieure à la moyenne. D'autre part, ce mode est plus spécialement destiné aux compositeurs ou improvisateurs qui aimeraient bien souvent se rappeler exactement ce qu'ils ont joué dans la minute qui précède plus tôt que ce qu'ils ont joué un quart d'heure avant. Le problème de saturation de la file d'entrée sera donc résolu dans ce cas si on convient de prendre une file d'entrée cyclique, l'interprétation par le calculateur n'étant lancée que lorsque l'utilisateur le demandera.

III - 3 - d) MODE 4

On entre sous le mode par une commande de la forme :

MISAJOUR titre, sous-titre, m

où m indique le numéro de la voix (fichier L.I.) ou désigne le fichier L.S.

Les effets de cette commande ne sont pas bien définis actuellement, les fonctions autorisées sous le mode 4 (pour les fichiers L.I.) seront vraisemblablement de la forme :

- supprimer une fonction non argument d'une autre fonction
- intercaler une fonction de L.I.
- ...

De toute façon ces fonctions ne posent pas de problèmes musicaux supplémentaires.

Pour complément d'informations voir le chapitre IX.

CHAPITRE IV : LE LANGAGE INTERNE (L.I.)

Bon nombre de langages de représentation interne de la musique ont été définis à ce jour, mais il n'existe pas, à vrai dire de langage universel. Par suite plutôt que d'utiliser un de ces langages nous avons préféré en définir un plus particulièrement adapté à notre système. Le langage qui semble cependant actuellement le plus utilisé est le langage DARMS défini par Stefan BAUER-MENGELBERG [5] [9] et il serait donc intéressant d'étudier le passage de L.I. à DARMS dans le but d'utiliser éventuellement des programmes d'analyse déjà existants si, par exemple, une analyse de style est envisagée pour des partitions écrites en L.I.

IV - § 1) GENERALITES

IV - 1 - a) POURQUOI L.I.

Les langages de représentation interne de la musique ont en général été étudiés pour permettre une analyse sémantique de la musique, c'est-à-dire une description des "événements musicaux" d'une partition, sachant que nous appelons "événement musical" un son et sa durée ou un silence et sa durée. En outre peuvent être spécifiés la clé, le ton et la mesure d'une partition ainsi que certains caractères spéciaux comme le point d'orgue ou la trille, mais il ne sera pas possible de donner beaucoup d'indications sur le graphisme même de tous ces concepts, comme par exemple les coordonnées des caractères spéciaux ou le nombre d'espaces entre les notes. L.I. ne prétend pas être sensiblement différent de ces langages quant aux concepts qui peuvent être représentés mais, et c'est là la différence essentielle, jusqu'à présent une partition à introduire en mémoire était codée à la main alors que le moyen d'entrée que nous proposons consiste en un clavier de piano et un certain nombre de pédales et touches de fonctions ; en regard de ce dispositif, L.I. a été défini de telle sorte qu'il permette une traduction aisée des signaux d'entrée.

IV - 1 - b) PRINCIPALES INDICATIONS D'ESTHETIQUE

Il est évident que, pour une partition créée sous le mode 3, aucune indication quant à l'esthétique de la partition éditée ne pourra être donnée par l'utilisateur (sauf éventuellement lors de la mise à jour ultérieure). C'est surtout sous le mode 2 (mode destiné aux éditeurs) que des indications pourront être données. Les principales sont les suivantes :

- indication d'espace horizontal entre les notes, silences, barres de mesure, etc....,
- indication des coordonnées exactes des commentaires, silences, caractères spéciaux, etc....,
- indication du sens des barres de notes,
- indication de mise automatique à l'octave.

Lorsqu'aucune indication n'est donnée (en particulier sous le mode 3) une option par défaut est prise.

IV - 1 - c) L.I. : LANGAGE DE FONCTIONS

Considérons l'exemple (à éditer) suivant :



Pour pouvoir imprimer le premier Do, il faut attendre que le Mi soit donné en entrée, car la longueur des barres des notes Do, Ré et Mi et la longueur de la barre commune dépendent des trois notes Do, Ré et Mi. La fonction "CRO" (croche) a donc été créée, le nombre de ses arguments est variable et ceux-ci peuvent être des notes, des silences ou des caractères spéciaux ; toutes les notes faisant partie de ses arguments auront une barre horizontale commune. "CRO (" est la traduction en L.I. du signal correspondant à une pression sur la pédale "croche" (cf chap. V) ; le relâchement de la pédale étant traduit en L.I. par ")". Toute fonction entrée et traduite en L.I. entre la pression et le relâchement de la pédale est considérée comme argument de la fonction "CRO".

De même "début de triolet" est traduit en L.I. par "NLT 3 (" et "fin de triolet" est traduit par ")".

La traduction de l'exemple précédent en L.I. est :

NLT 3 (CRO (NOT (40) NOT (42) NOT (44)) CRO (NOT (45) NOT (47) NOT (49))
CRO (NOT (51) NOT (52) NOT (54)))

Cette notation fonctionnelle correspond assez bien au style des informations d'entrée (" pression pédale" et "relâchement pédale", "début triolet" et "fin triolet") et il a finalement été convenu de considérer tous les éléments de L.I. comme des fonctions à nombre variable d'arguments, à un argument (par exemple la fonction "NOT") ou sans argument.

IV - § 2) DEFINITION FORMELLE

L'ensemble des symboles ou fonctions du langage L.I. peut se diviser en 5 groupes : le premier regroupe les fonctions sans argument ou à un argument simple (un nombre ou une chaîne de caractères) ; le deuxième regroupe les fonctions à nombre variable d'arguments, les arguments pouvant être des fonctions du langage ; le troisième regroupe les fonctions à un argument qui peut être une fonction du langage ; le quatrième et le cinquième regroupent des fonctions spéciales.

Notion de pointeur : tout au long de la traduction (sous forme de partition) d'une chaîne L.I. deux pointeurs (un horizontal et un vertical) pointent vers l'emplacement où doit être placé le premier symbole de la prochaine fonction à traduire. Le déplacement des pointeurs peut être provoqué au moyen de touches du clavier de fonctions ; le déplacement du pointeur horizontal est, d'autre part, provoqué implicitement après la mise en place d'un symbole. Nous noterons PH et PV les valeurs des pointeurs horizontal et vertical. Voir le paragraphe f) pour plus de détails.

N.B. : pour une bonne compréhension des fonctions, des exemples sont donnés dans le paragraphe g). Pour les fonctions soulignées dans les paragraphes a) à e) voir le paragraphe f).

IV - 2 - a) GROUPE I

Ce sont les fonctions :

<u>SIL</u>	<u>NDG</u>	<u>T18</u>	<u>T19</u>	<u>T20</u>	<u>T21</u>	<u>T22</u>
<u>T23</u>	<u>T24</u>	<u>T25</u>	<u>T26</u>	<u>T27</u>	<u>T30</u>	<u>T31</u>
<u>T32</u>	<u>VRO</u>	<u>VBL</u>	<u>VNO</u>	<u>VCR</u>	<u>VDC</u>	<u>VTC</u>
<u>VQC</u>	BAM	CLE	TON	MES	<u>CHA</u>	NOT

Les fonctions SIL et NDG sont des fonctions à un argument. L'argument de la fonction SIL est un chiffre de 1 à 7 ; la fonction SIL étant la fonction "silence", le chiffre 1 indique une pause, le chiffre 2 une demi-pause, le chiffre 3 un soupir, etc... (voir ex. 2 au paragraphe g)). La fonction NDG est la fonction doigté, son argument peut être un chiffre de 0 à 5 ou une des lettres p, i, m, ou a ; cet argument est le symbole de doigté que l'on veut éditer. Les symboles spécifiés par les fonctions SIL et NDG seront édités aux emplacements précisés par les PH et PV courants.

Les fonctions T18 à T31 sont des fonctions sans argument correspondant respectivement aux caractères musicaux spéciaux de la liste suivante :

q % n m k ~ s ^ v • Δ ⊕ ⌘

Les fonctions VRO à VQC sont des fonctions sans argument correspondant respectivement aux symboles suivants :

○ ♩ ♪ ♫ ♬ ♭

Ces dernières fonctions ne servent pas à imprimer les notes sur la portée, mais sont utilisées pour l'impression des commentaires donnant la durée réelle de l'unité de temps (ex. : ♩ = 120). L'emplacement des symboles est celui donné par les PH et PV courants.

La fonction BAM est une fonction sans argument et représentant une barre de mesure.

Les fonctions CLE, TON et MES sont des fonctions à un argument représentant un changement de "clé" ou "ton" ou une nouvelle indication de mesure. L'argument est un nombre et précise la nouvelle clé, le nouveau ton ou la nouvelle indication de mesure (voir ex. 1 au paragraphe g)).

La fonction CHA est une fonction à un argument qui est une chaîne de caractère à imprimer en commentaire (voir annexe III). L'emplacement est celui précisé par les

PH et PV courants.

La fonction NOT est une fonction à un argument. La fonction indique qu'une note est à éditer, l'argument donne sa hauteur.

IV - 2 - b) GROUPE II

Ce sont les fonctions :

REL RON BLA NOI CRO DCR TCR QCR NLT ACC ACA

La fonction CRO a été vue au § 1) c) ; les fonctions RON, BLA, NOI, DCR, TCR et QCR sont générées suivant le même principe mais indiquent respectivement des rondes, blanches, noires, doubles croches, triples croches et quadruples croches.

La fonction REL est également générée à l'aide d'une pédale et permet d'indiquer que toutes les notes jouées entre l'appui et le relâchement de la pédale doivent (si possible) avoir une barre commune.

La fonction NLT est, comme les précédentes, une fonction à nombre variable d'arguments, mais n'est pas générée au moyen d'une pédale puisqu'en fait il n'y aura pas qu'une mais plusieurs fonctions NLT :

NLT 3 → triolet

NLT 6 → sextolet

-

-

NLTn → n-olet

Ces fonctions indiquent que le chiffre n et une liaison reliant le premier et le dernier silence ou note faisant partie de leurs arguments doivent être édités. D'autre part il est possible de préciser si la liaison et le chiffre doivent être au-dessus ou au-dessous des silences et notes considérées. Il faudra donc dans ce cas préciser : NLTn H ou NLTn B. L'ensemble des fonctions NLT est donc l'ensemble :

$$\{ \text{NLTn, NLTn H, NLTn B} \mid n \in [2, +\infty[\cap \mathbb{N} \}$$

Une position pour la liaison et le chiffre sera évidemment prise par défaut si ni H ni B n'est précisé.

Nous avons préféré introduire une infinité de fonctions NLT dans la définition mnémonique de L.I. mais, en mémoire, il n'y en aura qu'une seule, le chiffre et la position seront précisés par des arguments supplémentaires. De même, en entrée sous le mode 2, "début NLT" et "fin NLT" seront générés au moyen de touches de fonctions, le chiffre et la position seront précisés au clavier alphanumérique.

Les fonctions ACC et ACA indiquent que les silences et notes faisant partie de leurs arguments doivent être édités avec la même valeur de PH. ACC signifie "accord", ACA signifie "accord arpégé" ; ces deux fonctions sont générées au moyen de pédales. (Voir ex. 17 au paragraphe g)).

Remarque : la fonction CRO peut ne pas avoir d'argument NOT. Pour la signification de ce cas particulier voir les exemples 8 et suivants du paragraphe g).

IV - 2 - c) GROUPE III

Ce sont les fonctions :

BEH BEB APP PAC PAX ADI ADD ABE ADB ABC PAN

Les fonctions BEH et BEB signifient que le calcul, déterminant le sens des barres des notes apparaissant dans la fonction argument de BEH ou BEB, ne doit pas être effectué et que les barres doivent être dirigées (respectivement) vers le haut ou vers le bas.

Les fonctions APP, PAC et PAX signifient que toutes les notes, apparaissant dans la fonction argument de APP, PAC ou PAX, sont (respectivement) des appoggiatures, des notes carrées ou des notes en croix.

Les fonctions ADI, ADD, ABE, ADB et ABC signifient que le calcul de l'altération logique éventuelle, à mettre devant les notes apparaissant dans la fonction argument d'une de ces fonctions, ne doit pas être effectué et que l'on impose à ces altérations d'être (respectivement) un dièze, un double dièze, un bémol, double bémol ou un bécarre :

ex. NOT (41) correspond au do # ou au ré b . Si le calcul de l'altération logique entraîne l'édition d'un do # mais que l'on veuille éditer un ré b il faudra préciser ABE (NOT (41)).

NOT (40) correspond au do. Supposons que le calcul de l'altération logique implique l'édition d'un do, ADI (NOT (40)) provoquera l'édition d'un si \sharp . Mais, par contre, ABE (NOT (40)) n'a aucun sens car NOT (40) ne peut être qu'un do, un si \sharp ou un ré $\flat\flat$. Dans ce cas c'est quand même l'altération (éventuelle) logique qui sera prise.

La fonction PAN signifie que pour toutes les notes, apparaissant dans la fonction argument de PAN, la mise automatique (et éventuelle) à l'octave des notes trop hautes ou trop basses ne doit pas être effectuée et que ces notes doivent être éditées avec le nombre de lignes supplémentaires nécessaire.

IV - 2 - d) GROUPE IV

Ce sont les fonctions :

X HPT LIA

La fonction X est une fonction à un argument qui est un nombre ($\in \mathbb{N}$) et signifie que le pointeur horizontal doit être incrémenté de la valeur du nombre. Bien qu'étant une fonction simple, cette fonction n'est pas mise dans le groupe I, car elle pose des problèmes à l'élaboration en vue de l'édition si le nombre est négatif.

La fonction HPT signifie que tous les silences et notes de la fonction précédente doivent être pointés (voir ex. 16 au paragraphe g)).

La fonction LIA signifie que toutes les notes apparaissant dans la fonction précédente doivent être reliées avec les notes correspondantes de même hauteur les précédant. Seules sont signalées, au moyen de la fonction LIA, les liaisons rythmiques c'est-à-dire celles influant sur la durée des sons. Les liaisons mélodiques sont décrites dans le paragraphe e).

N.B. : le fait que la fonction LIA s'applique aux notes de la fonction précédente et aux notes les précédant est pratique pour l'utilisateur. Pour simplifier la programmation il est préférable que la fonction LIA s'applique aux notes de la fonction précédente et aux notes les suivant. La décision sera à prendre lors du rodage du prototype ; actuellement, la deuxième solution est prise en compte (voir ex. 15 au paragraphe g)).

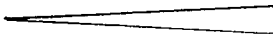
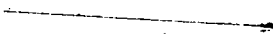
IV - 2 - e) GROUPE V

Ce sont les fonctions :

DBC FNC DLM FLM

Ces fonctions n'ont pas d'argument.

Les fonctions DBC et FNC signifient respectivement "début crescendo" et "fin crescendo".

DBC..... FNC génèrera 
FNC DBC génèrera 

Les positions du début et de la fin des crescendos sont celles des PH et PV courants lorsqu'apparaissent les fonctions DBC et FNC. La longueur des crescendos est donc $|PH_{DBC} - PH_{FNC}|$.

Les fonctions DLM et FLM signifient "début (resp. ^{nt} fin) de liaison mélodique". Les positions des extrémités de la liaison sont celles des PH et PV courants lorsqu'apparaissent DLM et FLM.

IV - 2 - f) PH et PV

Dans une chaîne L.I., pour toute fonction non argument d'un autre fonction la valeur d'une variable appelée PHA (fonction du PH) sera précisée pour faciliter, en particulier, la mise à jour.

ex. CRO (NOT (40) NOT (42)) CRO (RON (NOT (40)) NOT (42))

sera correctement écrit comme suit :

CRO. valeur de PHA. (NOT (40) NOT (42)) CRO. valeur de PHA.(RON(NOT(40))
NOT (42))

Ces valeurs de PHA seront calculées et insérées par le programme au moment de la traduction en L.S. (des précisions sont données dans le chap. VII - 4 - e).

Il n'est pas nécessaire de connaître les valeurs du PH courant avant la traduction en L.S.

.../...

Par contre, tout au long de la création des chaînes de L.I. le PV courant doit être accessible. En effet, pour les fonctions soulignées dans les paragraphes précédents, les valeurs du PV courant doivent être connues au moment de la traduction en L.S. et ne peuvent pas être calculées par le programme de traduction puisque l'édition d'un symbole ne provoque pas de modification implicite du PV (contrairement au PH) et les variations explicites (au moyen de touches du clavier de fonctions) **ne sont pas**, comme pour le PH, traduites en L.I. sous forme de fonctions. Pour les fonctions soulignées la valeur du PV sera donc insérée par le programme au moment de la création de la chaîne L.I.

ex. CRO. valeur de PHA.(NOT(40) SIL.valeur de PV.(4) NOT(40) NOT (40))

Si l'on travaille sous le mode 3, les fonctions ne seront pas toutes utilisées (en particulier les fonctions représentant les symboles spéciaux) ; parmi les fonctions soulignées, seule la fonction SIL pourra être utilisée, mais sous le mode 3 le PV n'est pas accessible par l'utilisateur ; pour la fonction SIL une valeur par défaut du PV sera donc prise.

IV - 2 - g) EXEMPLES

Nous concrétiserons ce qui précède par une série d'exemples (voir en annexe II les résultats de simulation sur terminal I.B.M. 2741 muni d'une boule "plotter") mais, pour ne pas les surcharger, sans insérer les valeurs des PHA et PV. Dans tous les exemples on est en clé de sol et le ton est do (sauf pour le premier).

Notation en L.I.

Résultat à l'édition

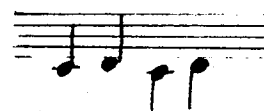
1 - CLE (1) TON (0.5) MES (2.2)



2 - CRO(NOT(40)SIL(4)NOT(44))CRO(NOT(40)NOT(42)NOT(44))



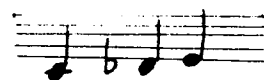
3 - NOI(NOT(40)NOT(42)) BEB(NOI(NOT(40) NOT(42)))



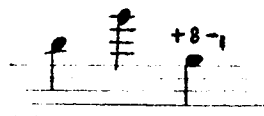
4 - NOI (NOT(40) NOT(41) NOT(42))



5 - NOI(NOT(40) ABE(NOT(41)) NOT(44))



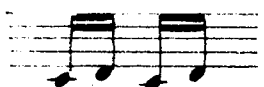
6 - NOI(NOT(63) PAN(NOT(71)) NOT(71))



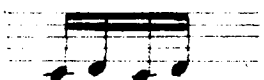
7 - APP(CRO(NOT(40)NOT(42)NOT(44)))NOI(NOT(45))



8 - DCR(NOT(40)NOT(42))DCR(NOT(40)NOT(42))



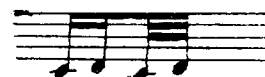
9 - REL(DCR(NOT(40)NOT(42))DCR(NOT(40)NOT(42)))



10 - REL(DCR(NOT(40)NOT(42)) TCR(NOT(40)NOT(42)))



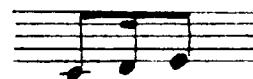
11 - REL(DCR(NOT(40)NOT(42))CRO()TCR(NOT(40)NOT(42)))



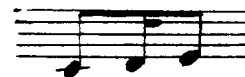
12 - REL(DCR(NOT(42)NOT(44)) NOI(NOT(42))CRO(NOT(44))
DCR(NOT(42))CRO()TCR(NOT(44))QCR(NOT(42))
DCR(NOT(44))CRO()DCR(NOT(42)))



13 - REL(CRO(NOT(40))DCR(NOT(42))CRO(NOT(44)))



14 - REL(CRO(NOT(40))CRO()DCR(NOT(42))CRO(NOT(44)))



15 - CRO(NOT(40)NOT(42)LIA NOT(44))CRO(NOT(42))



16 - NOI(NOT(42) NOT(42) HPT) HPT



17 - CRO(ACA(NOI(NOT(42)) NOT(45)NOT(52))NOT(47))
NOI(ACC(NOT(40) NOT(45)))



Remarque : toutes les manipulations possibles (sous le mode 2) des claviers (clavier de piano, clavier alphanumérique et clavier de fonctions) et des pédales sont à priori autorisées. Une manipulation anarchique entraîne la création d'une chaîne de L.I. peut-être syntaxiquement correcte mais pleine d'ambiguïtés ou de non sens du point de vue de l'édition. Nous avons entrevu ces problèmes dans le paragraphe c) au sujet des altérations. D'où la nécessité d'opérer, sur les chaînes de L.I., un prétraitement pour éliminer les aberrations (et les signaler éventuellement à l'utilisateur) avant la traduction en L.S. La liste des interdictions est donnée dans le chapitre V.

IV - § 3) REPRESENTATION EN MEMOIRE

La solution évidente pour représenter une chaîne de L.I. est de faire correspondre aux caractères alphanumériques leur code-caractère et aux valeurs numériques leur valeur en binaire mais il est préférable d'étudier une autre solution par souci d'économie de mémoire. Le système de représentation suivant est un de ceux pouvant être proposé pour une machine à octets, l'étude de rentabilité n'étant pas exhaustive puisque la solution définitive dépendra du calculateur utilisé dans le système prototype.

L'élément de base de mémoire est donc l'octet. Seront codés au moyen d'octets :

- les types de fonctions,
- les parenthèses fermantes des fonctions à plusieurs arguments,
- les valeurs des PHA et PV,
- les arguments terminaux (sachant que nous appelons arguments terminaux les arguments des fonctions à un argument du groupe I et celui de la fonction X.)

Ne seront pas codés :

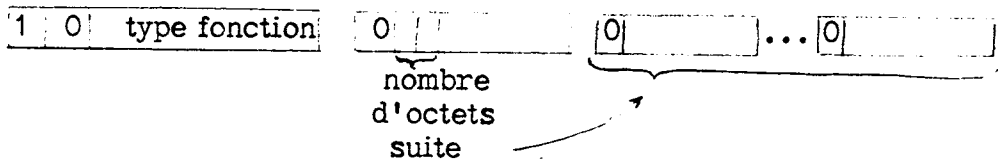
- les parenthèses ouvrantes de toutes les fonctions,
- les parenthèses fermantes des fonctions à un argument,
- les points délimitant les indications de PHA et de PV

Le principe de représentation en mémoire est tel que l'analyse sémantique du contenu des octets ne pourra pas être faite récursivement. Pour permettre cependant une analyse que nous appellerons simili-réursive (en particulier pour faciliter les recherches dans les chaînes d'octets lors de la mise à jour), il est convenu que les seuls octets, dont les deux bits de gauche sont 10, sont des octets contenant, dans les 6 bits restant, le code d'une fonction argument d'aucune autre fonction dans la chaîne L.I. (ceci est possible puisque L.I. ne compte que 57 fonctions). D'autre part, les seuls octets contenant 11 dans les deux bits de gauche sont des octets contenant, dans les 6 bits restant, une indication de PV (qui pourra donc être comprise entre - 32 et + 32).

Tous les autres codes seront à 7 bits, le bit de gauche de l'octet étant à 0.

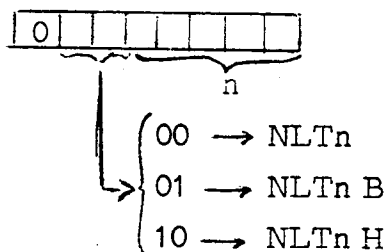
Problèmes particuliers :

- une indication de PHA suit toujours un octet du type 10 xxxxxx et utilisera 1, 2, 3 ou 4 octets :

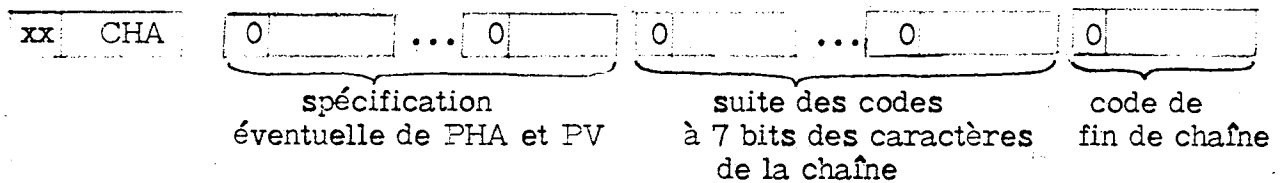


- pour les arguments terminaux, un ou deux octets sont nécessaires mais le nombre est fixe pour un type de fonction donné.

- l'ensemble des fonctions NLT ne sera représenté en mémoire que par un seul "type fonction" (cf § 2) b)) ; d'où la nécessité d'utiliser un argument supplémentaire qui sera situé immédiatement après la spécification éventuelle de PHA. Le contenu de cet octet est le suivant :



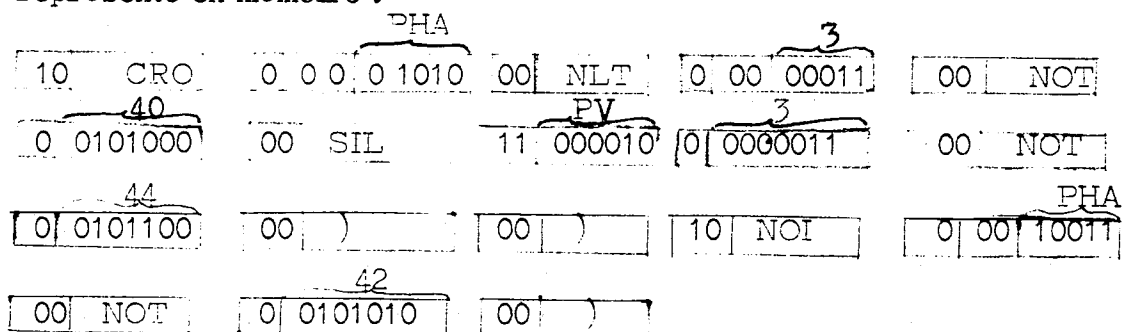
- Fonction CHA :



- C'est pour ne pas rendre obligatoire la spécification de PV et, de ce fait, ne pas se trouver face à des ambiguïtés lors de l'analyse d'une chaîne, qu'il a été nécessaire de définir une marque caractéristique des octets PV (la marque 11 dans les deux bits de gauche).

N.B. : la marque caractéristique 10 des octets contenant les types des fonctions non arguments d'autres fonctions permettra un retour-arrière dans la chaîne mais, pour éviter les ambiguïtés, l'analyse des fonctions ne pourra avoir lieu que de la gauche vers la droite.

ex. CRO.10.(NLT 3(NOT(40)SIL.2.(3)NOT(44)))NOI.19(NOT(42))
représente en mémoire :

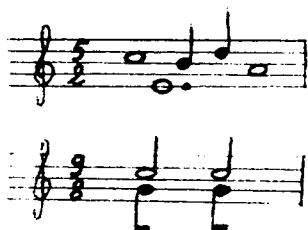


IV - § 4) COMPATIBILITE AVEC DARMS

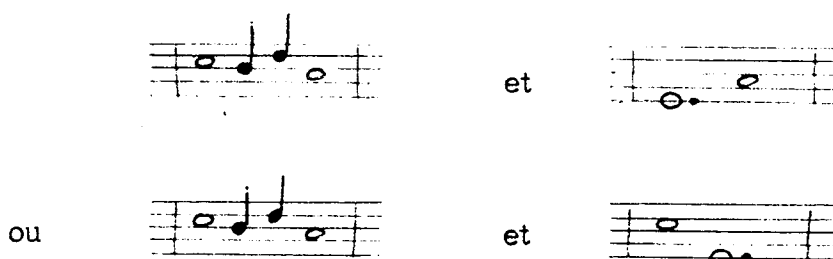
DARMS étant un langage assez couramment utilisé pour l'analyse sémantique de la musique il serait intéressant d'étudier les problèmes de passage de L.I. à DARMS en vue de faire de l'analyse sémantique des partitions (créées à l'aide de notre système) en utilisant les programmes d'analyse existants, et d'étudier les problèmes de passage de DARMS à L.I. pour pouvoir éditer des partitions écrites en DARMS.

Le passage de DARMS à L.I. ne semble pas poser de problèmes épineux puisqu'à une partition bien définie en DARMS il correspond toujours au moins une partition dans l'écriture musicale habituelle et, si peu d'indications pour l'édition sont indiquées en DARMS, on pourra prendre un certain nombre d'options par défaut.

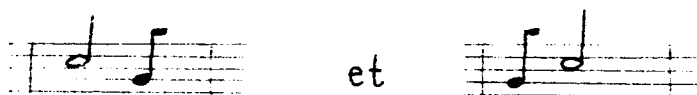
La passage de L.I. à DARMS pose, par contre, des problèmes puisque certaines notations correctes en L.I. peuvent ne pas avoir de signification musicale réelle ou être sujettes à des ambiguïtés comme dans les exemples suivants :



Le premier exemple est-il la réunion des voix :



Le deuxième exemple est-il erroné ou est-il la réunion des deux voix suivantes ? :



Il s'agit ici, évidemment, de cas rares car l'utilisateur s'appliquera en général à créer des partitions facilement lisibles donc facilement interprétables par programme mais plus on voudra rester libéral quant aux notations autorisées plus les programmes de passage de L.I. à DARMS devront être sophistiqués.

Si l'utilisateur veut bien respecter une certaine discipline il sera facile de lui faire accepter certaines contraintes d'écriture à définir lors de l'écriture des programmes de passage de DARMS à L.I. Bien sûr ces contraintes ne seront pas imposées aux gens préoccupés uniquement par l'édition, car les problèmes d'analyse sémantique (donc de traduction en DARMS) ne les intéressent pas.

CHAPITRE V : PASSAGE DES SIGNAUX D'ENTREE A L.I. ET A L.I. PRETRAITE

V - § 1) CORRESPONDANCE ENTRE LES INFORMATIONS D'ENTREE ET L.I.

Les informations d'entrée, sous les modes 2 et 3, seront codées, par le programme de traitement des interruptions et stockées en séquence dans une file d'entrée en vue du traitement par les programmes d'édition.

V - 1 - a) MODE 2

Ici, les informations d'entrée proviennent des claviers et des pédales et seront stockées, dans la file d'entrée, sous la forme de suites de bits correspondant à la représentation en mémoire d'éléments du langage L.I. (cf chap. IV - 3). Pour une meilleure compréhension nous ne donnerons pas ici la représentation en mémoire mais le code mnémonique L.I. des informations introduites dans la file d'entrée.

Le clavier alphanumérique est en général bloqué et ne sera activé qu'à la suite de l'appui sur une des touches de fonctions 5, 6, 7, 14, 15, 16 et 50 (voir annexe IV) pour permettre la frappe de l'information complémentaire ; le "retour-chariot" entraînera le reblocage du clavier alphanumérique ; rien ne sera introduit dans la file d'entrée après l'appui sur une de ces touches de fonctions mais uniquement après le "retour-chariot" :

- pour la touche n° 7, NL_{Tn}H(ou NL_{Tn}B(
(suivant l'information complémentaire) sera introduit dans la file d'entrée,

- pour les autres touches (5, 6, 14, 15, 16 et 50),
code (n) sera introduit dans la file d'entrée, code étant le code mnémonique de la touche de fonction considérée, n étant le code L.I. de l'information complémentaire.

L'appui sur les touches de fonctions 2 et 4 provoquera respectivement la mise dans la file d'entrée de X(1) et X(-1).

L'appui sur les touches de fonctions 1 et 3 provoquera l'incréméntation ou la décrémentation d'un compteur (initialement à 0). La valeur de ce compteur sera prise et insérée dans la file d'entrée chaque fois qu'une fonction de L.I., pour laquelle le PV doit être précisé, sera mise dans la file d'entrée.

L'appui sur la touche de fonction 8 provoquera la mise dans la file d'entrée de).

L'appui sur la touche de fonction 51 provoquera la mise dans la file d'entrée d'un "code-fonction" spécial ne faisant pas partie du langage L.I. et indiquant que la fonction achevée ou inachevée précédente (voir "fonctions achevées" au chap. VII) doit être ignorée. Le code spécial et la fonction seront effacés lors d'un premier traitement avant l'édition.

L'appui sur une des touches de fonctions non citées ici et données en annexe IV provoquera la mise dans la file d'entrée de l'élément de L.I. de même nom que le code mnémonique de la touche considérée.

L'appui sur les pédales provoquera respectivement la mise dans la file d'entrée de REL(, RON(, BLA(, NOI(, CRO(, DCR(, TCR(, QCR(, ACC(et ACA(.

Le relâchement d'une pédale provoquera la mise dans la file d'entrée de).

L'appui sur une des touches du clavier de piano provoquera la mise dans la file d'entrée de NOT(n), n étant le numéro correspondant à la touche. Le relâchement de la touche sera ignoré.

Remarque 1 : voir remarque du § b)

Remarque 2 : il faut éliminer les combinaisons du genre de la suivante :

RON(... ACC(... BLA(... NOI(...)) ... CRO(...))...)

.../...

Les fonctions concernées sont les fonctions du groupe II (chap. IV - 2 - b). Pour éviter ces combinaisons erronées, l'algorithme suivant est proposé :

- pour chaque signal de début de fonction le nom de la fonction est mis en tête d'une pile ;
- pour un signal de fin de fonction détecté, deux cas se présentent :
 - . le nom de la fonction est dans la pile, dans ce cas tous les noms éventuels situés au-dessus et le nom considéré sont effacés, autant de symboles) sont introduits dans la file d'entrée ;
 - . le nom de la fonction n'est pas dans la pile, dans ce cas le signal est ignoré.

V - 1 - b) MODE 3

V - 1 - b - 1 - File d'entrée

Sous le mode 3, les seules informations d'entrée sont celles provenant du clavier de piano et du métronome électrique. Ces informations doivent être traduites en L.I. sous forme de notes et silences avec leurs durées.

L'appui sur une touche du piano provoque la mise dans la file d'entrée du numéro de la note et de la valeur de l'horloge (cf chap. III - 3 - c), de même pour le relâchement de la touche et de même pour chaque top de métronome si l'on convient de considérer que le métronome a un numéro particulier (évidemment le même pour chaque top).

Remarque : pour éviter d'être obligé de mettre une marque de distinction entre les appuis et les relâchements de note il faudra veiller, lors de l'initialisation du mode 3, à ce que toutes les touches du piano soient en position relevée pour être sûr que le premier signal envoyé par chaque note corresponde bien à un appui sur la touche. Même remarque pour les notes et les pédales sous le mode 2.

Si le métronome n'est pas d'une extrême précision, il est préférable, comme nous le proposons, d'introduire l'heure dans la file d'entrée à chaque top pour maintenir un bon synchronisme entre l'utilisateur et le rythme donné au départ.

V - 1 - b - 2 - Passage à L.I.

Le passage à L.I. pose ici plus de problèmes que sous le mode 2, car si la hauteur des notes est facilement repérable, il n'en est pas de même pour le calcul des durées des notes et des silences. En théorie la solution est simple : on calcule $n = (\text{heure de relâchement} - \text{heure d'appui}) / \text{durée de base d'une noire}$; si $n = 1$, c'est une noire ; si $n = 2$, c'est une blanche ; si $n = 1/2$, c'est une croche ; etc... De même pour les silences : on peut facilement calculer la durée des temps morts que l'on divisera par la durée de base d'un soupir (= durée de base d'une noire). Mais la pratique montre que n a peu de chances d'être un entier ou l'inverse d'un entier, dans le meilleur des cas il en sera très voisin. Nous avons préféré mettre en évidence ces problèmes de rythme dans un chapitre réunissant les problèmes épineux (chap.IX) car les problèmes d'approximations ou estimations des durées des notes jouées sont loins d'être aussi simples qu'on pourrait le penser.

V - § 2) PASSAGE DE L.I. A L.I. PRETRAITE

Nous avons vu, dans le chap. IV - 2 - c à propos des altérations que, si l'utilisateur ne respecte pas une certaine discipline, des aberrations peuvent être introduites dans les chaînes de L.I. Pour laisser cependant à l'utilisateur une certaine liberté d'action, il est nécessaire de ne pas interdire toute manipulation superflue ou contradictoire, mais de supprimer, par programme, les aberrations en les signalant éventuellement à l'utilisateur (par le biais de l'imprimante), surtout si des choix arbitraires ou des suppressions ont été effectuées par le programme dans des cas douteux.

Pour faciliter la traduction en L.S., il a donc été prévu dans un premier temps de passer par un langage intermédiaire que nous appellerons "L.I. prétraité" et qui est un sous-ensemble de L.I. dans lequel sera traduite la chaîne d'entrée purgée de ses aberrations et simplifiée.

Trois traitements principaux sont opérés sur les chaînes d'entrée :

- suppression des fonctions à annuler (celles qui sont suivies du code spécial d'annulation) ;

- premier traitement sur les notes et altérations ;

.../...

- suppression des aberrations, contradictions et superfluités ; simplification.

V - 2 - a) TRAITEMENT SUR LES NOTES ET ALTERATION S

V - 2 - a - 1 - Généralités

Lors de la création d'une voix d'un morceau de musique, la clé et le ton sont mis en mémoire et restent accessibles tout au long de la création ; les valeurs seront modifiées par les fonctions CLE et TON du langage L.I.

Si les notes du piano sont numérotées de 1 à 85 (en fait le prototype prévu n'en comportera que 61), pour chaque note on peut avoir plusieurs noms possibles :

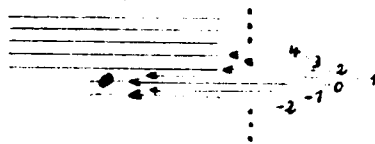
ex. : - la 40e note (le do du milieu du piano) pourra être un do naturel, un si # ou un ré bb ,

- la 41e note pourra être un do# ou un ré b ,

- la 42e note pourra être un ré naturel, un do x ou un mi bb ,

- etc...

La connaissance du ton courant nous permettra de choisir le nom le plus logique à donner, la connaissance de la clé nous permettra de déterminer la place sur la portée. Les notes sur la portée seront numérotées (quelle que soit la clé) comme suit :



ex.: le ton est do# , la clé est fa 4e, la note jouée est la 42e note du piano ; par suite, le nom de la note est do x et les différentes représentations sont les suivantes:

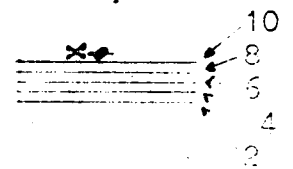
en L.I.

en L.I. prétraité

sur la portée

NOT(42)

ADD(NOT(12))



.../...

Remarque : les fonctions "altérations" en L.I. ne servent qu'à imposer un nom à une note parmi les différents noms possibles (cf chap. IV-2-c), les fonctions "altérations" en L.I. prétraité spécifient les altérations effectives à éditer.

V - 2 - a - 2 - Algorithmes

Pour la recherche de l'altération éventuelle et du numéro de la note blanche du piano à altérer correspondant à la note jouée, nous nous limiterons aux notes (du piano) n° 40 à 51, le problème pour les autres notes étant le même à une transposition près.

Dans les algorithmes suivants les notes 40 à 51 seront numérotées de 1 à 12, les notes blanches de 1 à 7.

Le principe utilisé pour la détermination du nom des notes est le suivant : si le ton est do, les notes 1 à 12 auront pour noms : do, do \sharp , ré, mi \flat , mi, fa, fa \sharp , sol, sol \sharp , la, si \flat , si ; pour les autres tons, la notation utilisée est une transposition de celle-ci.

N.B. : Dans ce qui suit des opérations modulo 7 seront faites sur les chiffres de 1 à 7 et des opérations modulo 12 sur les chiffres de 1 à 12. Bien noter que le chiffre 0 n'est pas employé, c'est-à-dire que $7 + 1 = 1 \pmod{7}$, $7 + 2 = 2 \pmod{7}$, etc... et $12 + 1 = 1 \pmod{12}$, $12 + 2 = 2 \pmod{12}$, etc...

Les algorithmes sont donnés en pseudo-PL/1.

V - 2 - a - 2a - Algorithme de changement de ton

A chaque changement de ton un tableau d'entiers T(12,2) est réinitialisé de la manière suivante :

```
T(2,1), T(4,1), T(7,1), T(9,1), T(11,1) = 0 ;  
T(1,1) = 1 ; T(3,1) = 2 ; T(5,1) = 3 ; T(6,1) = 4 ;  
T(8,1) = 5, T(10,1) = 6 ; T(12,1) = 7 ;  
IF pas d'altération à la clé THEN Do ; J = 11 ; GOTO SUITE ; END ;
```

.../...

```

IF pas de dièze à la clé THEN GOTO BEMOL ;
Do i = 1 TO 7 ;
    IF un dièze est à la clé pour la note i (mod. 7)
    THEN Do ; chercher la valeur J comprise entre 1 et 12 telle que  $T(J, 1) = i$  ;
         $T(\frac{J+1}{\text{mod.12}}, 2) = i$  ; END ;
END ; GOTO SUITE ;
BEMOL : J = 0 ;
Do i = 1 TO 7 ;
    IF un bémol est à la clé pour la note i (mod. 7)
    THEN Do ; K = J ; chercher la valeur J comprise entre 1 et 12 telle que  $T(J, 1) = i$  ;
         $T(\frac{J-1}{\text{mod.12}}, 2) = i$  ; END ;
END ; J =  $\frac{J-3}{\text{mod.12}}$  ; IF K = 0 THEN J = 6 ;
SUITE :  $T(\frac{J+3}{\text{mod.12}}, 2)$ ,  $T(\frac{J+8}{\text{mod.12}}, 2)$ ,  $T(\frac{J+10}{\text{mod.12}}, 2) = 0$  ;
 $T(\frac{J+5}{\text{mod.12}}, 2)$ ,  $T(\frac{J+12}{\text{mod.12}}, 2) = -1$  ;
pour tout K tel que  $T(K, 2)$  n'est pas encore affecté faire  $T(K, 2) = T(K, 1)$  ;

```

V - 2 - a - 2b - Algorithme de détermination de la note et de son altération logique.

L'algorithme suivant détermine la note (mod. 7) à mettre sur la portée et l'altération éventuelle à mettre devant (les altérations à la clé ne sont évidemment pas éditées).

Quelle que soit la note i (mod. 12) jouée, la note (mod. 7) à éditer est :

```

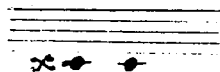
IF  $T(i, 2) > 0$  THEN  $T(i, 2)$  ;
ELSE IF  $T(i, 2) = 0$  THEN
    IF  $T(\frac{i-1}{\text{mod.12}}, 1) > 0$  THEN  $\# T(\frac{i-1}{\text{mod.12}}, 1)$  ;
    ELSE IF  $T(i, 1) = T(\frac{i-1}{\text{mod.12}}, 2)$  THEN  $\flat T(i, 1)$  ;
    ELSE  $\% T(\frac{i-2}{\text{mod.12}}, 1)$  ;
    .../...

```


Si les deux notes sont dans la même mesure l'interprète lira en fait deux mi \flat
L'édition correcte serait :



Le \flat n'est pas vraiment une altération de passage mais une altération détruisant l'effet de la précédente. De même si le ton est do \sharp , l'édition de do ~~\sharp~~ ne sera pas



mais



(On pourrait convenir de n'éditer, pour le deuxième do, qu'un \sharp non précédé de \flat , mais l'usage veut que le \flat soit explicitement indiqué).

D'où l'algorithme suivant, servant à déterminer les altérations éventuelles à mettre pour détruire l'effet des précédentes :

A tout changement de ton un tableau TAB1 (1 : 7) est rempli avec les altérations du nouveau ton :

ex. ton la \Rightarrow 3 \sharp à la clé (fa, do et sol)

TAB1 (1) = \sharp , TAB1 (2) = rien, TAB1 (3) = rien, TAB1 (4) = \sharp ,

TAB1 (5) = \sharp , TAB1 (6) = rien, TAB1 (7) = rien

"rien" signifie absence d'altération.

Les éléments de TAB1 ne peuvent contenir qu'une des trois valeurs : \sharp , \flat , ou "rien".

Un tableau TAB2 (1 : 7) contient les altérations de passage et est mis à jour chaque fois que l'on a traité une note de la chaîne L.I.

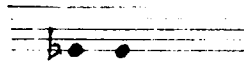
Appelons ALT (i) l'altération déterminée au moyen des algorithmes 2a et 2b pour une note i (mod. 7). Appelons EDI (i) l'altération à éditer. Le tableau suivant donne la valeur de EDI (i) et la nouvelle valeur de TAB2 (i) en fonction de ALT (i) et de la valeur courante de TAB2 (i) :

ALT (i)	TAB2 (i) (valeur courante)	EDI (i)	TAB2 (i) (nouvelle valeur)
b	{ rien b # x b bb }	b	b
rien	rien	rien	rien
rien	b	TAB1 (i)	rien
rien	{ # x b bb }	b TAB1 (i)	rien
#	{ rien b # }	#	#
#	{ x b bb }	b #	#
b	{ rien b b }	b	b
b	{ bb # x }	bb	b
x	{ rien b # x }	x	x
x	{ b bb }	b x	x
bb	{ rien b b bb }	bb	bb
bb	{ # x }	b bb	bb

Remarque : A ce stade, il n'y aura plus d'erreurs d'interprétation possibles, mais nous pourrions compléter ces algorithmes pour améliorer l'édition du point de vue esthétique. En effet, au moyen de ces algorithmes deux mi b consécutifs seront édités comme suit :



mais, pour alléger l'écriture, si ces deux notes sont dans la même mesure il faudrait éditer :

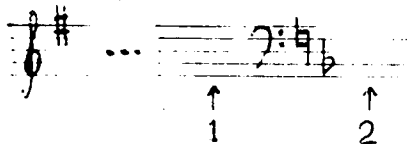


Ceci est possible, à condition de pouvoir déterminer, pour les notes altérées telles que $ALT(i) = TAB2(i)$, si la même note (mod. 7) précédente est ou non dans la même mesure. Le calcul est possible, mais pas simple du fait que les barres de mesure ne sont, en général, indiquées que dans une des voix. Il faut donc prévoir de compter les temps au cours de l'élaboration des voix ou d'indiquer les barres de mesure dans toutes les voix ; sinon il faudra se résigner à répéter les altérations.

V - 2 - a - 3 - Problème du "retour-arrière".

La fonction X de L.I. peut, en théorie, admettre un argument (entier) positif, négatif ou nul (si l'argument est nul la fonction sera ignorée). Un argument négatif ne sera en fait autorisé que dans certains cas (voir chap. VII - 1 - b - 3). En particulier un argument négatif pour la fonction X sera interdit après un changement de ton ou de clé.

ex.



Si, jusqu'à ce qu'on arrive à la position 2, il n'y a pas, dans la chaîne L.I., de fonction X à argument négatif, une note éditée en position 2 aura une position sur la portée et une altération calculée en fonction de la clé de fa et du ton si b . Si alors une fonction X apparaît dans la chaîne L.I. avec un argument négatif provoquant un retour à la position 1, une note éditée en position 1 aura une position sur la portée et une altération calculée toujours en fonction de la clé de fa et du ton si b alors

que les calculs devraient être effectués en fonction de la clé de sol et du ton sol.

V - 2 - b) RESTRICTIONS DIFFERENCIANT L.I. PRETRAITE DE L.I.

- REL non argument de RON, BLA, ..., QCR, ACC ou ACA
- BEH ou BEB toujours en tête d'une fonction achevée.
- Pas plus de deux fonctions (parmi RON, BLA, ..., QCR) actives en même temps.
 - La fonction NOT apparaît au moins une fois dans les fonctions RON, BLA, NOI, DCR, TCR et QCR (NOT peut ne pas apparaître dans les arguments de CRO : cf chap. IV - 2 - g - 11).
 - ACC ne peut pas apparaître dans les arguments de ACA et inversement.
 - X et NLT ne peuvent pas apparaître dans les arguments de ACC ou ACA
 - ACC et ACA apparaissent toujours dans les arguments d'une et d'une seule fonction RON, BLA, ..., ou QCR.
 - Parmi les fonctions RON, BLA, ... et QCR l'apparition d'une seule fonction (et une seule apparition) peut avoir lieu dans les fonctions ACC et ACA et un seul argument NOT peut apparaître dans les arguments de cette fonction.
 - HPT suit toujours NOT (...), SIL (...) ou HPT ; LIA suit toujours NOT (...) ou HPT.
 - ADI, ADD, ABE, ADB, ABC, APP, PAN, PAC, PAK sont toujours regroupés devant les fonctions NOT (PAC et PAK seront ignorés en présence de la fonction APP).
 - Si PAN ou APP apparaissent dans les arguments d'une fonction ACC ou ACA mettre PAN (resp. ^{nt}APP) devant toutes les fonctions NOT apparaissant dans la fonction ACC ou ACA.
 - Des fonctions K consécutives sont traduites en une seule, l'argument résultant étant égal à la somme des arguments de ces fonctions consécutives.
 - K (n) (n < 0) ne peut pas apparaître dans les arguments d'une autre fonction.

Après un premier traitement sur les notes et les altérations, les chaînes de L.I. sont transformées de telle sorte qu'elles répondent aux restrictions ci-dessus. Ces restrictions n'impliquent cependant pas que les chaînes obtenues soient éditables : en particulier dans le cas où K (n) (n < 0) est autorisé, n ne pourra pas être inférieur

à une certaine valeur, mais la valeur limite dépend de la configuration de la chaîne et sera déterminée lors du passage de L.I. prétraité à L.S. (chap. VII)

Les algorithmes permettant de satisfaire à ces restrictions n'ont pas été étudiés en priorité puisque ne posant pas de problèmes d'ordre musical.

V - 2 - c) DIFFERENCE ENTRE LE MODE 2 ET LE MODE 3

Ces problèmes de remodelage des chaînes L.I. en chaînes de L.I. prétraité n'apparaîtront pas sous le mode 3, car aucune indication sur le type et la forme des fonctions de L.I. à utiliser ne peut être donnée explicitement par l'utilisateur sous le mode 3. Les programmes, travaillant sur des signaux d'entrée dont le nombre de types est limité, pourront donc donner aux chaînes de L.I. des formes les plus simples possibles, c'est-à-dire facilement éditables. Ceci ne veut pas dire que le mode 3 soit plus facile à programmer que le mode 2 car il ne faut pas perdre de vue les problèmes ardu de détermination des durées des notes (voir chap. III).



CHAPITRE VI : LE LANGAGE DE SORTIE (L.S.)

VI - § 1) CARACTERES MUSICAUX

L'organisation anarchique dans l'écriture de la musique nous a conduits à définir un ensemble de caractères de même taille nous permettant, en les assemblant, de reconstituer des partitions musicales comme un véritable puzzle (à la seule différence que les caractères peuvent être superposés et utilisés plusieurs fois). Pour de plus amples renseignements sur les caractères, voir le chapitre VIII et les annexes V et VI. Le moyen de description des assemblages correspondant à des partitions est le langage L.S.

VI - § 2) L.S.

Soit VM l'ensemble des p caractères musicaux choisis. Le principe de description consiste à indiquer, ligne après ligne, les caractères composant ces lignes. Si RC signifie : se positionner en début de ligne suivante ; si BS signifie : reculer d'un "espace-caractère" sur la ligne courante ; si K (n) signifie : avancer de n "espace-caractère" sur la ligne courante ; et si C (m) désigne le même caractère de l'ensemble VM, alors les chaînes L.S. représentant les partitions sont de la forme :

$$[RC^* \quad K(n)^* \quad C(m)^+ \quad (BS \quad C(m)^*)^*]^*$$

étant entendu que :

1°) $(\dots K(n)^* \dots)^*$ représente

$$\underbrace{(\dots K(n_{11}) \quad K(n_{12}) \quad K(n_{13}) \dots, \dots)}_{*} \quad \underbrace{(\dots K(n_{21}) \quad K(n_{22}) \quad K(n_{23}) \dots, \dots)}_{*} \dots$$

*

Même remarque pour C (m).

2°) K (n) sous-entend $n \in \{1, 2, \dots, 99\}$

C (m) sous-entend $m \in \{1, 2, \dots, 26\}$

.../...

3°) Le dernier caractère C (m) (compté à partir d'un RC jusqu'au RC suivant) ne doit pas avoir une position (sur la ligne) supérieure à la position 100.

N.B. la représentation en mémoire des chaînes L.S. dépendra du calculateur utilisé dans le système prototype et ne pose pas de problèmes particuliers quant à la recherche de la meilleure solution à prendre pour un encombrement minimal de la mémoire.

CHAPITRE VII : DE L.I. PRETRAITE A L.S.

VII - § 1) PRINCIPES GENERAUX

VII - 1 - a) TABEAU TRANSITOIRE

Le principe de base pour la traduction des chaînes de L.I. prétraité en L.S. consiste à utiliser un tableau intermédiaire (TAB) à trois dimensions dans lequel seront rangés les numéros des caractères musicaux à éditer. Les valeurs des deux premiers indices indiquent respectivement le PH relatif et le PV du caractère considéré. Si $TAB(i, j, 1) = 0$ il n'y aura qu'un caractère blanc à l'emplacement i, j . Si à l'emplacement i, j , il doit y avoir les caractères n° 43, 87 et 52 nous aurons :

$TAB(i, j, 1) = 43, TAB(i, j, 2) = 87, TAB(i, j, 3) = 52, TAB(i, j, 4) = 0$

Le PH est un PH relatif car les chaînes L.I. seront élaborées morceaux par morceaux, ces morceaux étant ce qu'on appelle "fonctions achevées" et sont définis dans le paragraphe b).

La taille d'un caractère musical de base est de 5×5 (largeur \times hauteur). Sachant que la hauteur est égale à la dimension d'un interligne de portée, les caractères musicaux sont donc tous de la taille d'une note (sans sa barre éventuelle). 100 caractères en largeur seront donc approximativement nécessaires pour représenter une portée d'une page de partition et 10 à 20 caractères en hauteur. Rares seront les fonctions achevées générant une portée tout entière mais ce cas n'est, en théorie, pas extrême. Et si l'on estime que jusqu'à 5 caractères pourront être imprimés au même endroit il nous faudra donc un tableau TAB de dimension $100 \times 20 \times 5 = 10\ 000$ octets = 10 K - octets (si tous les numéros des caractères peuvent être codés sur un octet). Cela nous conduit à prévoir un calculateur de taille mémoire supérieure à 10 K octets ; et que se passera-t-il pour les cas extrêmes où des fonctions achevées génèrent plus d'une portée ? On peut, simplement, détourner le problème si l'on constate qu'en fait, un tableau représentant un morceau de partition contiendra une

quantité très importante de caractères blancs et nous avons donc convenu de ne représenter en mémoire que les éléments significatifs du tableau sous forme de liste.

ex. : si le tableau ne contient que les caractères 43 et 44 à l'emplacement i_1, j_1 et les caractères 12, 37 et 24 à l'emplacement i_2, j_2 le tableau sera représenté comme suit :

43	i_1	j_1	1	44	i_1	j_1	2	12	i_2	j_2	1	37	i_2	j_2	2	24	i_2	j_2	3
----	-------	-------	---	----	-------	-------	---	----	-------	-------	---	----	-------	-------	---	----	-------	-------	---

Mais par souci de clarté dans ce qui suit nous conserverons la notation de tableau.

VII - 1 - b) PRINCIPES DE TRAITEMENT

VII - 1 - b - 1 - Généralités

Fonction achevée : on appelle fonction achevée une fonction de L.I. dont la construction (lors de la mise dans la file d'entrée) ou l'élaboration (traduction de L.I. en L.I. prétraité ou de L.I. prétraité en L.S.) est achevée, cette fonction n'étant argument d'aucune autre fonction. (N.B. : la notion de fonction achevée existe à 3 niveaux).

Le principe de traitement consiste à élaborer une à une les fonctions achevées dès qu'une fonction est achevée dans la file d'entrée, elle est traduite en L.I. prétraité puis traduite de telle sorte que le tableau TAB puisse être rempli (N.B. : la valeur 0 pour le premier indice du tableau correspond en général (voir paragraphe 4) au PH de début de la fonction achevée) ; la représentation en L.S. de la fonction achevée est déduite du contenu de TAB ; avant de traiter la fonction achevée suivante le contenu de TAB est, en général, effacé (voir paragraphe 4 -').

VII - 1 - b - 2 - Problèmes de LIA, DBC et FNC, DLM et FLM

Chaque fois qu'une fonction LIA est trouvée dans la chaîne L.I., on met dans une liste T1 la note concernée et le PH courant :

T1	note 1	PH 1
	note 2	PH 2
	note 3	PH 3
	...	

Chaque fois que l'on rencontre une note dans la chaîne L.I. on regarde si elle se trouve dans T1 ; si oui, on peut mettre dans TAB les caractères correspondant à la liaison puis effacer l'enregistrement correspondant dans T1.

Chaque fois qu'une fonction DLM est trouvée dans la chaîne L.I., le PH et le PV correspondants sont mis en tête d'une pile T2 :

T2	PH1	PV1
	PH2	PV2
	PH3	PV3
	...	

Lorsqu'une fonction FLM est trouvée, celle-ci est ignorée si T2 est vide, sinon on met dans TAB les caractères correspondant à la liaison, les PH et PV du début de la liaison étant ceux situés en tête de pile T2 ; puis la tête de pile est effacée.

Une pile T3 contient les informations relatives aux DBC et FNC :

T3	DBC	PH1	PV1
	DBC	PH2	PV2
	...		

ou

T3	FNC	PH1	PV1
	FNC	PH2	PV2
	...		

Lorsque, dans la chaîne L.I., on trouve un DBC ou un FNC :

- un DBC \Rightarrow si T3 est vide ou contient des DBC, alors on met en tête de pile le DBC, sinon un decrescendo est mis dans TAB (les PH et PV de fin étant ceux du DBC, les PH et PV de début étant ceux du FNC situé au fond de la pile) et le FNC du fond de la pile est effacé.

- un FNC \Rightarrow si T3 est vide ou contient des FNC, alors on met en tête de pile le FNC, sinon un crescendo est mis dans TAB (les PH et PV de fin étant ceux du FNC, les PH et PV de début étant ceux du DBC situé au fond de la pile) et le DBC du fond de la pile est effacé.

VII - 1 - b - 3 - Problèmes de la fonction X

Trois écueils principaux s'opposent à l'autorisation de X (n) (n < 0) :

- X (n) apparaît avant ou après un changement de clé ou ton (cf chap.

V - 2 - a - 3),

- X (n) apparaît avant ou après des notes mises à l'octave,

ex. : BEH (NOI (NOT (71) NOT (71))) X (-3) BEB (NOI (NOT (47)))



à la lecture, le 3ème sol (NOT (47))
sera interprété à l'octave

- X (n) est tel que le nouveau PH courant correspond à un premier indice de TAB négatif.

Pour éviter les 1er et 3ème écueils on interdit à la fonction X (n) de précéder ou suivre une fonction achevée dans laquelle un changement de clé ou ton apparaît et on impose à n d'être tel que le nouveau PH courant soit supérieur ou égal à celui du début de la fonction achevée précédente. La valeur de n prise en considération par les programmes sera donc max (n minimum autorisé, n donné par la fonction X). Le n minimum autorisé est calculé, à chaque apparition d'une fonction X (n).

Pour éviter le 2ème écueil, la solution actuellement prise consiste aussi à interdire X (n) (n < 0) avant ou après une fonction achevée contenant des notes mises à l'octave. Une solution plus avantageuse pourra être envisagée si l'on conserve pour chaque fonction achevée les PH de début et fin des zones mises à l'octave, tant qu'un retour en arrière est possible.

VII - 1 - b - 4 - Tests de passage à la fonction suivante

Un élément de mémoire PHTAB contient la valeur du PH correspondant à la valeur nulle du premier indice de TAB.

Lorsque le traitement d'une fonction est achevé (c'est-à-dire que la fonction a été traduite et mise dans TAB) on traduit le contenu de TAB en L.S. et on l'intercale dans la chaîne L.S. déjà existante pour pouvoir lancer l'affichage sur l'écran ; puis on effectue les opérations suivantes :

.../...

- si T1, T2, T3 sont vides et si X (n) (n < 0) ne suit pas la fonction achevée traitée, alors on effectue l'opération suivante :

* on efface le contenu de TAB, on donne à PHTAB la valeur du PH courant et on va traiter la fonction suivante ;

- si X (n) (n < 0) suit la fonction achevée traitée on calcule la valeur de n finalement prise et on affecte à n cette valeur ; si n est positif ou nul on effectue l'opération* sinon on va traiter la fonction suivant la fonction X (n) ;

- si T1 ou T2 ou T3 n'est pas vide on va traiter la fonction suivante.

Remarque : Avant d'effectuer ces tests le PH courant repère la position où devrait commencer la fonction suivant la fonction qui vient d'être traitée. Si X (n) (n < 0) suit la fonction traitée, cette valeur du PH courant est mise de côté et c'est cette valeur que reprendra PH après traitement de la fonction suivant X (n) (n < 0).

VII - § 2) ORGANISATION DU TRAITEMENT D'UNE FONCTION ACHEVEE

Le traitement d'une fonction achevée s'opère en quatre pas.

VII - 2 - a) 1er PAS : EXAMEN DE LA FONCTION ; TRAITEMENT DE TOUT CE QUI N'EST PAS RELATIF A UNE NOTE .

La position (PH début et PH fin) et le type (n) des n-lets sont déterminés. La position et le type (+ 8, + 2 x 8, ...) des indications éventuelles de mise à l'octave sont déterminés, les fonctions NOT arguments de PAN n'étant pas concernées. Les liaisons et crescendos sont traités comme indiqué dans le § 1) b) 2 - et si, dans la fonction traitée, des renseignements, suffisants pour l'édition de liaisons et crescendos, apparaissent, alors ceux-ci sont collectés. Tous les renseignements éventuels (n-lets, indicateurs d'octaves, liaisons et crescendos) sont mis de côté et ne seront traduits pour être mis dans TAB qu'après traitement du reste de la fonction.

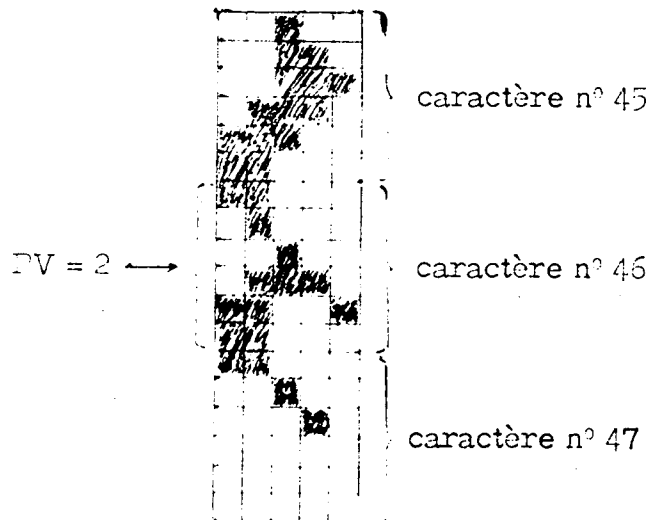
Toutes les informations relatives aux notes (REL, RON, BLA, ..., QCR, ACC, ACA, NOT et parenthèses de fin de ces fonctions (sauf NOT)) sont mises dans une chaîne appelée ITFONC. Le premier maillon de la chaîne est un indicateur de "barres en haut" ou "barres en bas" (BEH ou BEB) pour les notes de la fonction.

Les maillons NOT contiennent les renseignements (éventuels) : PAC, PAX, APP, altération, nombre de HPT, PH de la note et hauteur de la note sur la portée (calculée en tenant compte de la mise à l'octave éventuelle).

Tous les autres renseignements (fonctions SIL, CLE, TON, MES, doigtés et caractères spéciaux) sont traduits et mis dans TAB.

(Remarque : au sujet de la traduction de CLE et TON voir § 4 b.)

ex. : PHTAB = 112. On suppose que le PH courant à la valeur 115 et que l'on veut traiter SIL. - 2. (3) qui indique que l'on veut éditer un soupir ayant un PV égal à - 2. Le soupir se décompose en 3 caractères musicaux :



Après avoir cherché les plus petites valeurs (positives) de i_1 , i_2 et i_3 telles que $TAB(3, -3, i_1) = 0$, $TAB(3, -2, i_2) = 0$ et $TAB(3, -1, i_3) = 0$, nous ferons les affectations suivantes :

$$\begin{aligned}TAB(3, -3, i_1) &= 47 ; & TAB(3, -3, i_1 + 1) &= 0 ; \\TAB(3, -2, i_2) &= 46 ; & TAB(3, -2, i_2 + 1) &= 0 ; \\TAB(3, -1, i_3) &= 45 ; & TAB(3, -1, i_3 + 1) &= 0 ;\end{aligned}$$

VII - 2 - b) 2ème PAS : TRAITEMENT DES APPOGIATURES

La décision, quant au moyen de traiter les appoggiatures, n'a pas encore été prise. Deux possibilités sont envisagées :

- les traiter avant de traiter les autres notes (c'est à dire les traiter au 2ème pas) : dans ce cas, lors du traitement des autres notes, on pourra calculer des longueurs de barres de notes telles que les éventuelles barres communes des croches (doubles croches, ...) ne recoupent pas les appoggiatures ;

- les traiter après avoir traité les autres notes (c'est à dire au 3ème pas) : dans ce cas, on pourra placer les appoggiatures de telle sorte qu'elles ne se superposent pas aux altérations de la note ou de l'accord suivant.

Quelle que soit la possibilité choisie, deux solutions se présentent encore :

1 - une appoggiature se composant d'une seule note sera représentée par une croche barrée, une appoggiature se composant de plusieurs notes sera représentée par une série de double croches ayant une barre commune et ce quelles que soient les indications de durée (RON, BLA, ...) données pour les appoggiatures ;

2 - on tiendra compte des indications de durée pour les appoggiatures qui seront alors traitées comme les autres notes, à la seule différence qu'elles seront plus petites (et dans ce cas les fonctions APP seront traitées comme les fonctions PAC ou PAK).

VII - 2 - c) 3ème PAS : TRADUCTION DU CONTENU DE ITFONC RELATIF AUX NOTES NON APPOGGIATURES.

Si les appoggiatures sont traitées avant les autres notes, les renseignements relatifs aux appoggiatures seront effacés de ITFONC avant traitement des autres notes et inversement. On considérera ici que ITFONC ne contient que des renseignements sur les notes non appoggiatures.

VII - 2 - c - 1 - BEH ou BEB

Le premier maillon de ITFONC contient l'indication BEH, BEB ou rien. Si rien n'est précisé on recherche les plus haute et plus basse notes dans ITFONC en fonction desquelles on détermine alors le sens des barres des notes.

VII - 2 - c - 2 - Principe d'examen de la chaîne

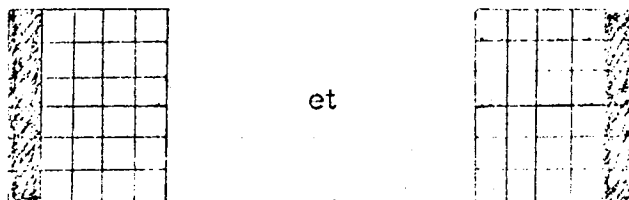
On parcourt, maillon par maillon, la chaîne ITFONC, en effectuant les opérations suivantes :

- Si on rencontre la fonction REL, on positionne un indicateur IREL.
- Si on rencontre une fonction durée de note, on positionne un indicateur ITYP ou ITYPP si ITYP est déjà positionné (c'est à dire qu'on a déjà rencontré une fonction durée). (Remarque : positionner ITYP (ou ITYPP) consiste à mettre le nom de la fonction durée dans ITYP (ou ITYPP)).
- Si on rencontre une fonction NOT :
 - si ITYP = RON, BLA ou NOI, on traduit et met dans TAB la note avec ses barrés) et altération éventuelles (si ITYPP contient un type la note a un deuxième type et donc éventuellement une deuxième barre en sens inverse de la première).
 - si ITYP = CRC, DCR, TCR ou CR on traduit la note et son altération éventuelle, sans sa barre principale mais avec sa deuxième barre éventuelle et on met le résultat dans TAB (voir § 3) c) 4-).
- Puis on met un enregistrement dans CHBAR NOT (voir paragraphe 3-).
- Si on rencontre une fonction ACC ou ACA on appelle un sous-programme ACCORD (voir paragraphe 4 -).
- Si on rencontre une parenthèse fermante, on efface le premier indicateur positionné rencontré en examinant dans l'ordre ITYPP, ITYP et IREL :
 - si ITYP a été effacé, IREL n'étant pas positionné ou si IREL a été effacé, on traduit et met dans TAB la chaîne CHBARNOT (voir paragraphe 7 -).
- Si on rencontre une marque de fin de la chaîne ITFONC, on arrête l'examen de la chaîne.

VII - 3 - c - 3 - CHBARNOT

CHBARNOT est une chaîne contenant des indications sur les barres de note des croches, doubles croches, triples croches et quadruples croches. Un maillon définit une barre de note : dans un maillon sont donc précisés le type (CRC, DCR, TCR ou CR), les III et IV du morceau de barre le plus proche de la note et la position de la barre au III considéré, c'est-à-dire "à droite" ou "à gauche" car nous disposons

de deux caractères pour éditer les barres de notes :



Lorsqu'on lance la traduction de CHBARNOT deux las se présentent :

- CHBARNOT n'a qu'un seul maillon,
- CHBARNOT a plusieurs maillons.

Si CHBARNOT n'a qu'un seul maillon, il s'agit de la barre d'une seule note :



Si CHBARNOT a plusieurs maillons toutes les barres seront reliées par une barre horizontale commune :

ex. :



VII - 2 - c - 4 - Sous-programme ACCORD

Le sous-programme ACCORD traite les maillons de la chaîne ITFONC compris entre celui représentant la fonction ACC (ou ACA) et celui représentant la parenthèse fermante de la fonction ACC (ou ACA). Les notes sont traduites avec leur deuxième barre éventuelle et mises dans TAB.

Si ITYP = RON, il n'y a pas de barre principale.

Si ITYP = BLA ou NOI, la barre principale est également traduite et mise dans TAB.

Si ITYP = CRO, DCR, TCR ou QCR, un enregistrement est mis dans TAB.

VII - 2 - d) 4ème PAS : n-OLETS, INDICATEURS D'OCTAVES, CRESCENDOS ET LIAISONS

Les indications de n-olets, d'octaves, les crescendos et les liaisons, mis de côté lors du 1er pas, sont traduits et mis dans TAB.

Remarque : l'ordre des pas est à respecter pour pouvoir obtenir une édition de bonne qualité. En effet, la longueur des barres de notes est calculée en fonction de la place disponible (dans TAB) de telle sorte que les barres des notes et les barres horizontales éventuelles ne se superposent pas aux caractères insérés lors du 1er pas. De même, il est préférable d'insérer en dernier dans TAB les indications de n-olets, d'octaves, les crescendos et les liaisons en fonction des places restant après insertion de toutes les barres de note.

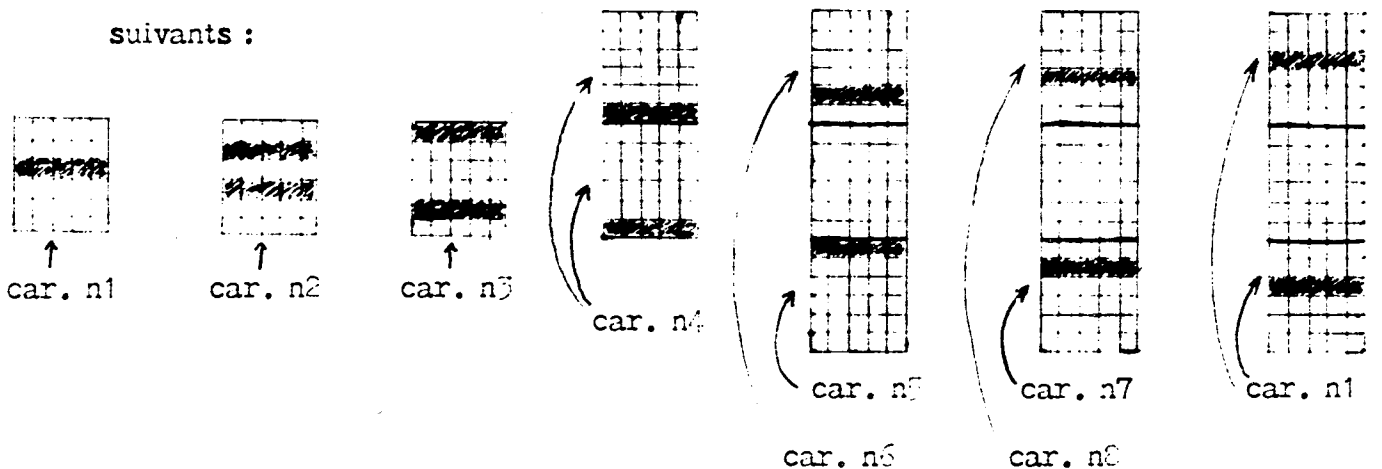
VII - § 3) ALGORITHMES PARTICULIERS

Il serait trop long de décrire ici tous les algorithmes nécessaires à la traduction d'une fonction achevée, nous décrivons les trois plus significatifs des problèmes à résoudre.

VII - 3 - a) ELABORATION DES CRESCENDOS

Nous prendrons le cas d'un crescendo dont l'axe de symétrie doit être situé à un FV égal à IPV, qui doit débuter à un PH égal à IPH1 et qui doit s'achever à un PH égal à IPH2.

Les caractères musicaux utilisés pour l'édition des crescendos sont les suivants :



Nous avons constaté que, dans les partitions musicales, on ne rencontre que rarement des crescendos dont l'écart à l'ouverture excède la taille de deux caractères, et ce quelle que soit la longueur des crescendos. Il a donc été convenu de n'utiliser, pour l'édition des crescendos, que les assemblages de caractères définis ci-dessus. Il est évident que, si des crescendos dépassent 7 caractères en longueur, certains assemblages devront être rejetés. Pour un choix harmonieux des assemblages à rejeter suivant la longueur des crescendos, le tableau ITCR (7,7) a été défini :

ITCR (1,1)							(1,7)
0	0	0	0	1	0	0	
0	0	0	1	0	1	0	
0	1	0	1	0	1	0	
1	0	1	0	1	0	1	
1	0	1	1	0	1	1	
0	1	1	1	1	1	1	
1	1	1	1	1	1	1	
(7,1)							(7,7)

L'algorithme suivant est utilisé pour traduire et mettre dans TAB les crescendos :

```

i = IPH2 - IPH1 ; N1 = FLOOR (i/7) ;
N2 = i - FLOOR (i/7) * 7 ; N = 1 ;
DO i = 1 TO 7 WHILE ( (N1 ≠ 0) | (i ≤ N2 + 1) ) ;
  IF N1 ≠ 0 THEN N = N1 + ITCR (N2 + 1, i) ; j = 0 ;
  DO WHILE (j < N) ; j = j + 1 ;
    parmi les instructions suivantes, exécuter l'instruction i :
    instr. 1 : mettre car. n1 dans TAB (IPH1 - PHTAB, IPV) ;
    instr. 2 : mettre car. n2 dans TAB (IPH1 - PHTAB, IPV) ;
    instr. 3 : mettre car. n3 dans TAB (IPH1 - PHTAB, IPV) ;
    instr. 4 : mettre car. n4 dans TAB (IPH1 - PHTAB, IPV) et TAB (IPH1 - PHTAB,
    IPV + 1) ;
  
```

```

instr. 5 : mettre car. n5 dans TAB (iPH1 - PHTAB, iPV - 1)
           et mettre car. n6 dans TAB (iPH1 - PHTAB, iPV + 1) ;
instr. 6 : mettre car. n7 dans TAB (iPH1 - PHTAB, iPV - 1)
           et mettre car. n8 dans TAB (iPH1 - PHTAB, iPV + 1) ;
instr. 7 : mettre car. n1 dans TAB(iPH1-PHTAB,iPV-1) et TAB(iPH1-PHTAB,
                                                         iPV + 1) ;

           iPH1 = iPH1 + 1 ;
END ;
END ;

```

Remarque : le problème des decrescendos est le même mais inversé.

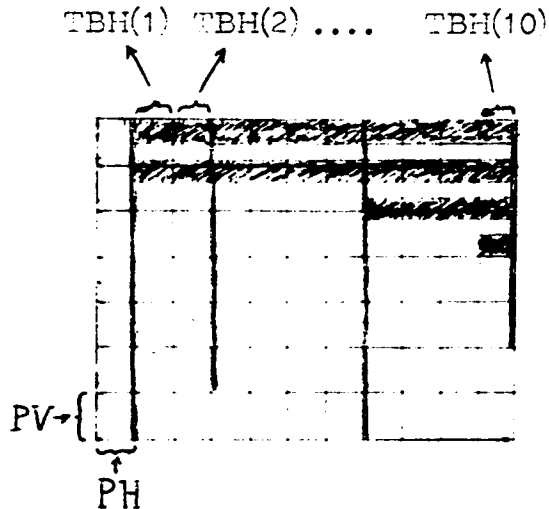
VII - 3 - b) ELABORATION DES BARRES DES NOTES

Nous n'étudierons pas ici les cas des blanches, noires ou ceux des croches (doubles croches, triples croches ou quadruples croches) solitaires car les problèmes sont triviaux ou découlent de ceux étudiés ici. Nous étudierons donc le traitement de CHBARNOT dans le cas où cette chaîne a plus d'un maillon.

Supposons que l'on ait à éditer 4 barres de notes dirigées vers le haut et spécifiées, dans CHBARNOT, au moyen des 4 maillons suivants :

DCR	PH	PV	D	DCR	PH+3	PV+1	G	TCR	PH+7	PV	G	QCR	PH+10	PV+2	D
-----	----	----	---	-----	------	------	---	-----	------	----	---	-----	-------	------	---

Le résultat à l'édition est de la forme :



VII - 3 - b - 1 - Calcul de la disposition des barres horizontales communes.

Nous nous proposons de mettre, dans chaque élément d'un tableau TBH (1 : n), le nombre de morceaux de barres horizontales devant apparaître à une position de PH donné (1 barre horizontale pour 1 croche, 2 barres pour une double croche, 3 barres pour une triple croche et 4 barres pour une quadruple croche).

Si nous notons CHBARNOT (i, j) le jème élément du ième maillon de CHBARNOT et si CHBARNOT (i, 4) est égal à 1 si la barre est à droite et à 0 si la barre est à gauche, alors la dimension n de TBH est égale à :

$(PH + 10) - PH + CHBARNOT (4,4) - CHBARNOT (1, 4)$ et l'élément TBH (1) correspond à la position :

$$PH + CHBARNOT (1, 4).$$

Plus généralement, si m est le nombre de maillons de CHBARNOT, la dimension de TBH est :

$$CHBARNOT (m, 2) - CHBARNOT (1, 2) + CHBARNOT (m, 4) - CHBARNOT (1, 4)$$

et l'élément TBH (1) correspond à la position :

$$CHBARNOT (1, 2) + CHBARNOT (1, 4)$$

Rajoutons TAB (0) = 0 et TAB (n+1) = 0, et supposons que CHBARNOT (i, 1) = 1, 2, 3, ou 4 s'il s'agit (respectivement) de CRC, DCR, TCR ou ICR.

L'algorithme de remplissage de TBH sera alors :

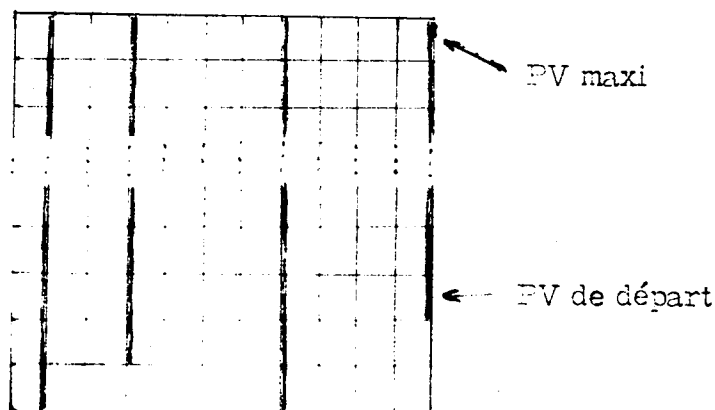
```
j = 1 ; Do I = 1 TO m-1 ; L = MIN (CHBARNOT (i, 1), CHBARNOT (i+1, 1)) ;
      Do K = CHBARNOT (i, 2) + CHBARNOT (i, 4)
      To CHBARNOT (i+1, 2) + CHBARNOT (i+1, 4) - 1 ;
      TBH (j) = L ; j = j+1 ; END ; END ;
Do I = 1 To m ;
  L = CHBARNOT (i, 2) - CHBARNOT (1, 2) + CHBARNOT (i, 4) - CHBARNOT (1, 4) ;
  IF MAX (TBH (L+1), TBH (L)) < CHBARNOT (i, 1)
  THEN IF TBH (L) >= TBH (L+1)
  THEN TBH (L) = CHBARNOT (i, 1) ;
  ELSE TBH (L+1) = CHBARNOT (i, 1) ;
END ;
```


VII - 3 - b - 2 - Recherche de la place disponible dans TAB

Du fait de l'autorisation (dans certains cas) de $n < 0$ dans la fonction $K(n)$ et, d'autre part, pour faciliter les opérations de mise à jour, nous nous proposons de faire une recherche de la place disponible dans TAB pour pouvoir caser les barres des notes sans recouper (si possible) d'autres caractères qui pourraient déjà s'y trouver.

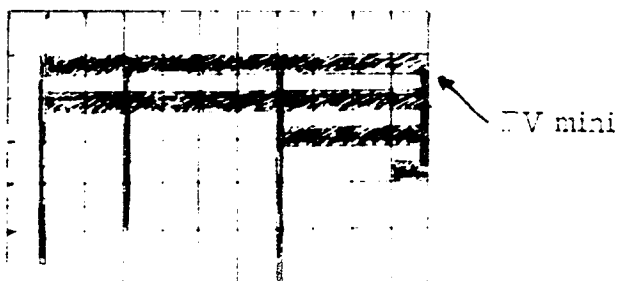
Nous ne donnerons pas ici les algorithmes en PL/I, une série de dessins seront plus explicites pour le lecteur. (Nous reprendrons pour cela l'exemple du début du paragraphe b)).

VII - 3 - b - 2a - Longueur maximale des barres (verticales) des notes



Sachant que les barres des notes ne peuvent recouper que des liaisons ou des lignes supplémentaires, on teste (en parallèle pour toutes les barres des notes) toutes les cases que peuvent traverser les barres des notes (à partir du PV le plus grand indiqué dans CHEARNOS jusqu'à une valeur limite) et l'on s'arrête dès qu'on atteint la valeur limite ou dès que l'on rencontre une case contenant un caractère musical ne servant pas pour les liaisons ou les lignes supplémentaires : (PV de cette case) - 1 est le PV maximum pour les barres des notes.

VII - 3 - b - 2b - Longueur minimale des barres (verticales) des notes

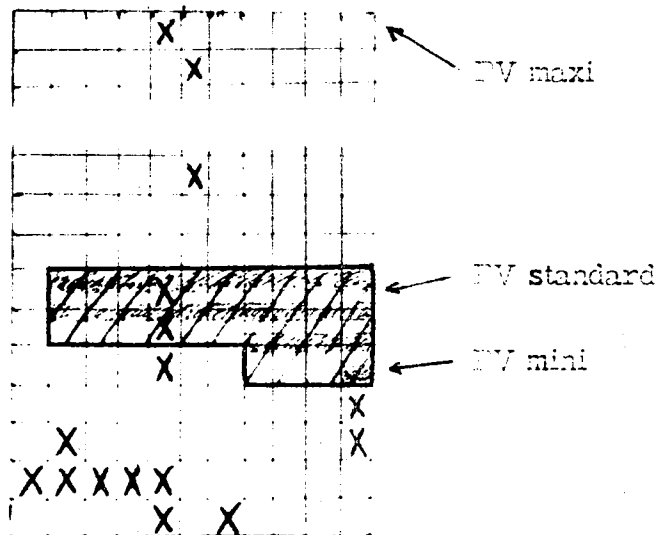


Le PV minimum pour les barres des notes n'est pas égal au PV le plus grand indiqué dans CHBARNOT car il faut prévoir de la place pour les barres horizontales. C'est en consultant parallèlement CHBARNOT et TBH que l'on peut calculer ce PV mini.

VII - 3 - b - 2c - Longueur "standard" des barres (verticales) des notes.

Le PV (du sommet des barres des notes) qui est choisi, quand la disponibilité des cases de TAB le permet, est égal à PV mini + p (p étant une constante ; actuellement nous avons pris $p = 2$).

VII - 3 - b - 2d - Recherche de la place disponible pour les barres horizontales.



La zone hachurée représente la zone recouverte par les caractères composant les barres horizontales des notes. A titre d'exemple les croix indiquent l'emplacement de caractères musicaux déjà mis dans TAB. La recherche consiste à déplacer cette zone, verticalement, à partir du PV standard jusqu'au PV maxi, puis du PV standard au PV mini, jusqu'à ce qu'aucune croix n'apparaisse dans la zone. Si le PV mini est atteint sans que l'on ait trouvé de place telle que la zone ne recouvre pas de croix, c'est le PV standard qui sera tout de même pris pour le sommet de la zone. Si $PV\ maxi < PV\ standard$, la recherche n'aura lieu qu'en faisant décroître PV maxi jusqu'à PV mini. PV standard sera aussi, dans ce cas, pris par défaut de place disponible.

Remarque 1 : le problème pour les barres (verticales) de notes dirigées vers le bas est le même que celui pour les barres dirigées vers le haut, mais inversé.

Remarque 2 : dans le § 3) b) ainsi que dans le § 2) c) 2-, il n'a pas été tenu compte des fonctions CRO sans argument NOT(cf chap. IV - 2 - g - 10 et 11) et ce, pour ne pas apporter, dans les algorithmes, des surcharges inutiles pour la compréhension des principes de base.

VII - 3 - c) ELABORATION DES ACCORDS

Nous considérons toujours ici que les barres (verticales) éventuelles sont dirigées vers le haut. Le problème pour les barres dirigées vers le bas est le même mais inversé : en fait les deux cas seront toujours élaborés par un même algorithme, la valeur de certains paramètres faisant la différence.

Nous n'avons pas introduit, jusqu'à présent, la différence fondamentale qui existe, du point de vue de l'édition, entre les notes situées sur les lignes de portée et celles situées entre les lignes ; car l'édition des notes ne faisant pas partie d'accords pose, à vrai dire, peu de problèmes. Mais nous devons introduire ici cette différence puisque l'élaboration des accords dépend de la disposition des notes sur la portée. Nous étudierons ici la fonction ACC, l'effet de la fonction ACA différant de l'effet de la fonction ACC par le seul fait qu'elle provoque l'édition, devant l'accord du symbole "arpège" : ce symbole a une longueur (en hauteur) égale à celle de l'accord et doit être placé juste à gauche de l'accord et de ses altérations éventuelles ; lors de l'élaboration de l'accord la place des altérations est déterminée et on pourra donc, en connaissance de cause, placer correctement le symbole.

VII - 3 - c - 1 - Forme des notes

- Prenons l'exemple d'une croche : 3 caractères musicaux sont utilisés pour l'édition :



ou



croche sur une ligne

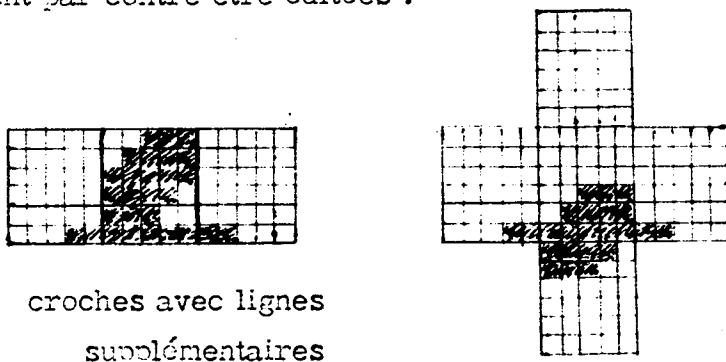
.../...

arc

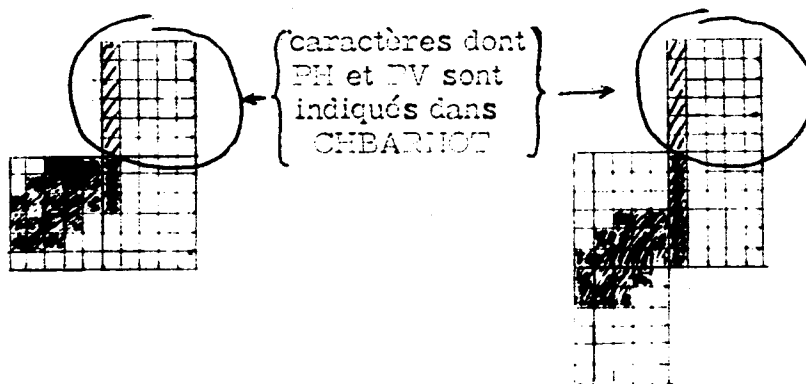
interligne

Les niveaux des lignes de portée sont ceux du bas des caractères.

- Les lignes de portée n'ont pas besoin d'être éditées par les programmes puisqu'elles seront générées automatiquement par hardware ; les lignes supplémentaires doivent par contre être éditées :



- Une croche ne faisant pas partie d'un accord est éditée comme suit :



- Pour les accords, des notes peuvent apparaître de chaque côté de la barre, donc il faut autoriser que les notes puissent chevaucher des morceaux de barre :



Etant entendu que sur l'écran de télévision on ne peut pas superposer de caractères il faut pour chaque superposition possible, définir un nouveau caractère. Pour limiter le nombre de caractères, il a été convenu que seules les notes situées dans des interlignes pourront être superposées aux barres. Il faudra donc disposer les notes en fonction de leur position par rapport aux lignes et de la position de la barre de note. L'élaboration des accords comprend l'élaboration de la barre de note jusqu'au niveau du caractère "note" le plus élevé, l'élaboration du reste de la barre étant ensuite faite par les programmes d'élaboration des barres des notes ne faisant pas partie d'accords.

Remarque : tout ce qui précède est valable pour les autres types de note, sauf pour les rondes en ce qui concerne les barres de note.

VII - 3 - c - 2 - Position des notes et de la barre de note

Soit NI la position de l'accord : appelons pos. 1 la position NI et pos. 2 la position $NI + 1$. Les notes vont se répartir entre pos. 1 et pos. 2, la barre sera à droite en pos. 1 ou à gauche en pos. 2. Les renseignements sur les notes sont pris en consultant $ITBENC$ (cf § 2 c) 1). Nous appelons "note paire" une note située sur une ligne et "note impaire" une note située dans un interligne (suite à la numérotation des notes indiquées dans le chap. V-2-a-1).

La manière de procéder pour l'élaboration des accords est la suivante :

- On cherche (dans $ITBENC$) la note la plus basse et la note la plus haute de l'accord ; soient (respectivement) NI et NI' les numéros de ces notes comme définis au chap. V-2-a-1 : cela nous permet de déterminer les deux ensembles de caractères possibles correspondant à la barre de note (le 1er ensemble contient des morceaux de barre "à droite" et sera à considérer si la barre doit être mise en pos. 1, le 2ème ensemble contient des morceaux de barre "à gauche" et sera à considérer si la barre doit être mise en pos. 2).

- Si NI est pair : placer les notes paires en pos. 1 ; placer le morceau de barre en pos. 1 (mettre dans TAB le 2ème ensemble de caractères) ; pour toute note impaire N : s'il n'existe pas de note paire N telle que $|N - NI| = 1$ alors placer N en

pos. 1 sinon placer K en pos. 2.

- Si H1 est impair : chercher la note paire la plus basse, soit H3 ; si H3 - H1 \neq 1, alors effectuer les mêmes opérations que si H1 était pair, sinon : placer les notes paires en pos. 2, placer le morceau de barre en pos. 1 (mettre dans TAB le 1er ensemble de caractères) et placer les notes impaires en pos. 1.

VII - 3 - c - 3 - Placement des notes

Dans le paragraphe 2 - ci-dessus, placer une note signifie : déterminer les caractères correspondants à mettre dans TAB ; le PH est connu (pos. 1 ou pos. 2), le PV des caractères est à calculer en fonction du numéro de la note. Il faut, pour cela, regarder (toujours dans ITFONC) le type principal de la note (ITYP) et son deuxième type éventuel (ITYPP).

En mémoire ITYP (ou ITYPP) = 1, 2, 3, 4, 5, 6, 7 ou 0 si ITYP (ou ITYPP) représente (respectivement) RON, BLA, NOI, GRO, DCR, TCR, LCR ou rien.

Sachant cela, on effectue les opérations suivantes :

- Si ITYPP = 0, une note de type ITYP, sans sa barre, est placée dans TAB.
- Si ITYPP \neq 0, une note de type MIN (ITYP, ITYPP) est placée dans TAB sans la barre correspondant à ITYP mais avec la deuxième barre éventuelle (correspondant à ITYPP) qui sera placée sur la gauche (et à l'extérieur) de la note et dirigée vers le bas.

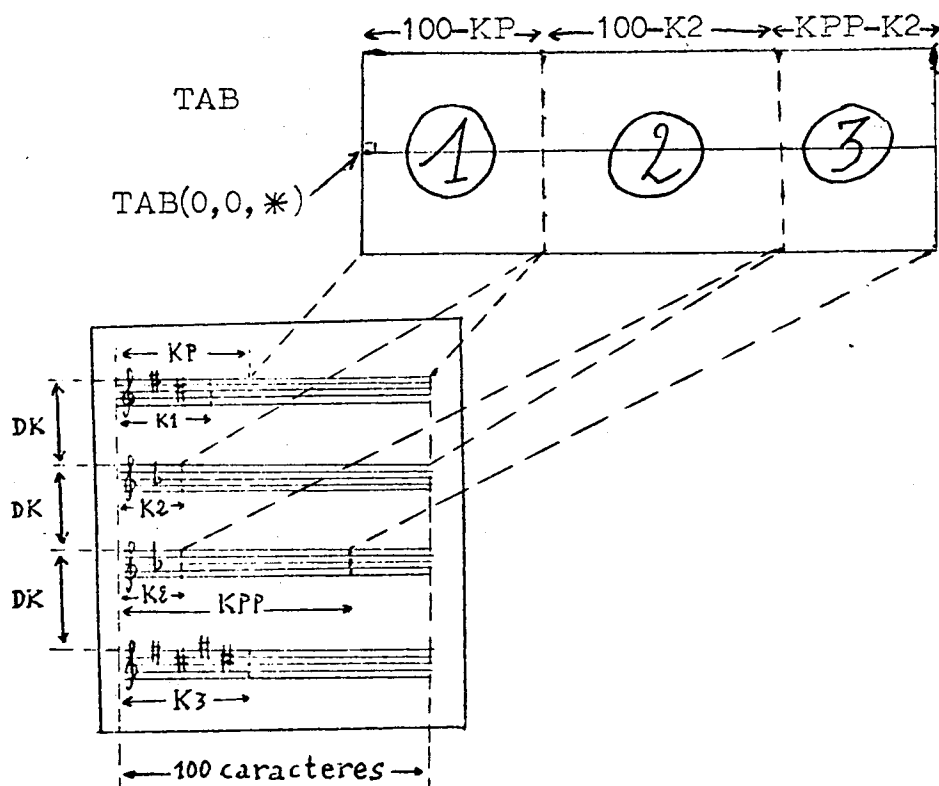
Remarque : chaque fois qu'une note est placée on regarde si une altération est à mettre : si oui, on cherche (en allant vers la gauche à partir de pos. 1 - 1) une place libre dans TAB et on met l'altération (une place libre est une zone pouvant contenir l'altération et ne contenant pas de caractères faisant partie d'autres altérations déjà placées).

VII - § 4) DANS TAB A L'É.

VII - 1 - a) GENERALITES

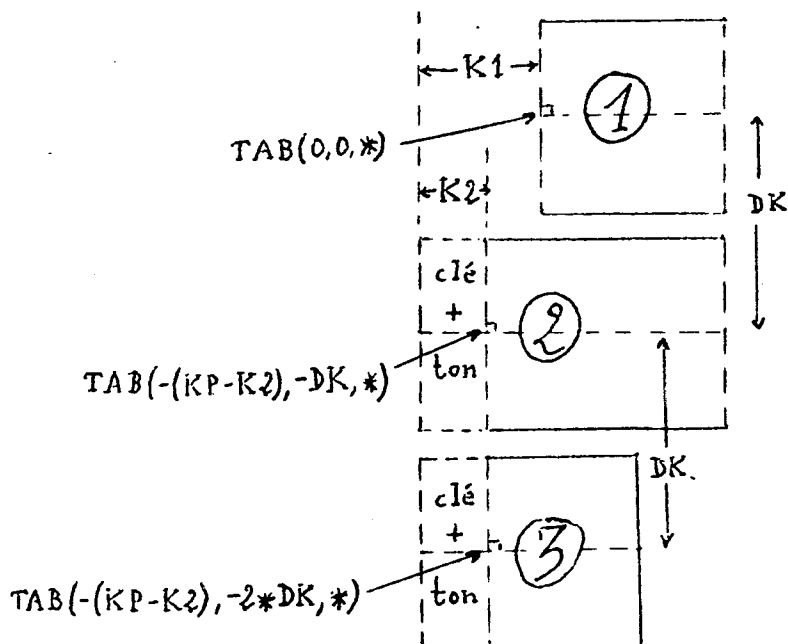
Le schéma suivant montre les relations existant entre la grille de caractères

qu'est une partition et la grille de caractères qu'est TAB :



DK indique une hauteur en caractères et est fonction du nombre de voix (l'exemple ci-dessus est à une voix). Toutes les autres distances sont des largeurs en caractères. $KP + 1$ est la position sur la ligne de partition courante à laquelle correspond $TAB(0, *, *)$; au moment de la traduction en L.S. d'une fonction achevée, KP doit être connu.

Avant la traduction en L.S. il est nécessaire de remanier TAB comme suit :



Les changements de clés et tons sont indiqués explicitement dans la chaîne L.I. et sont donc dans TAB, mais les clés et tons à mettre au début des lignes des partitions ne sont pas indiqués dans la chaîne L.I., il faut donc, lors du remaniement de TAB, les traduire et les mettre dans TAB.

VII - 4 - b) COMPLEMENTS SUR LE PASSAGE DE L.I. A TAB POUR LES FONCTIONS CLE ET TON

CLE (n), TON (n), K (n), I (n) sont des tableaux dont les valeurs sont mises à jour chaque fois que l'on traduit (et met dans TAB) une fonction CLE ou TON. Ces tableaux sont évidemment initialisés lors du lancement de la traduction d'une chaîne L.I. et doivent contenir (au moment où l'on entame la traduction d'une fonction achevée) les valeurs suivantes :

- CLE (1) contient la clé courante,
- TON (1) contient le ton courant,
- K(1) contient le nombre de caractères (en largeur) nécessités pour l'édition de la clé et du ton du début de la ligne courante de la partition,

- I (1) est la valeur du 1er indice de TAB correspondant à la position K (1) + 1 sur la ligne courante de la partition, cette valeur est donc, en général négative,
- CLE (2) = CLE (1),
- TON (2) = TON (1),
- K (2) contient le nombre de caractères (en largeur) nécessités pour l'édition de la clé et du ton courants (CLE (1) et TON (1)),
- I (2) = I (1) + 100 - K (1),
- les autres valeurs des tableaux sont à 0.

Traitement de la fonction CLE :

Lorsqu'on rencontre dans la chaîne L.I. une fonction CLE, on calcule I1 et I2 qui sont les valeurs du 1er indice de TAB la plus petite et (respectivement) la plus grande parmi les éléments de TAB devant contenir les éléments de la clé. Puis on cherche le plus petit J tel que CLE (J) = 0. On effectue alors les opérations suivantes :

si $I1 < I(J-1)$ alors $\left\{ \begin{array}{l} \text{CLE (J-1) = nouvelle clé ;} \\ \text{K (J-1) = \{nombre de caractères (en largeur) nécessités pour} \\ \text{l'édition de la clé CLE (J-1) et du ton TON (J-1) ;} \\ \text{J = J - 1 ;} \end{array} \right.$

sinon $\left\{ \begin{array}{l} \text{CLE (J) = nouvelle clé ;} \\ \text{TON (J) = TON (J - 1) ;} \\ \text{K (J) = \{nombre de caractères (en largeur) nécessités pour} \\ \text{l'édition de la clé CLE (J) et du ton TON (J) ;} \\ \text{I (J) = I (J-1) + 100 - K (J-1) ;} \\ \text{tant que } I1 \geq I(J) \text{ faire } I(J) = I(J) + 100 - K (J - 1) ; \end{array} \right.$

si $I2 \geq I(J)$ alors $\left\{ \begin{array}{l} \text{ne pas mettre dans TAB les caractères de la clé mais} \\ \text{incrémenter I1 de } I2 - I1 + 1 ; \end{array} \right.$

sinon mettre dans TAB les caractères de la clé.

Remarque : pour le traitement de la fonction TON, l'algorithme est symétrique à celui-ci.

VII - 4 - c) REMANIEMENT DE TAB

VII - 4 - c - 1 - Réorganisation du contenu

Pour ne pas modifier le contenu de I (n) on utilise un tableau II (n) (II (*) = I (*)).
L'algorithme est le suivant :
IDPT = 100 - K (2) ; N = 2 ; LL = 1 ;
ici : IF il n'y a pas d'élément $\neq 0$ dans TAB (JJ, *, *) pour JJ \geq II (N)
THEN GOTO FIN ;
Do L = II (N) To II (N) + 100 - K (N) - 1 ;
Do pour tout J tel que TAB (L, J, *) $\neq 0$;
TAB (L-IDPT, J - DK*LL, *) = TAB (L, J, *) ; TAB (L, J, *) = 0 ;
END ; END ;
IF II (N) + 100 - K (N) \neq II (N + 1)
THEN II (N) = II (N) + 100 - K (N) ; ELSE N = N + 1 ;
LL = LL + 1 ; IDPT = IDPT + 100 - K (N) ; GOTO ici ;
FIN : ...

VII - 4 - c - 2 - Insertion des clés et tons des débuts de ligne de partition

On suppose que le sous-programme d'insertion dans TAB des clés et tons de passage a besoin qu'on lui donne en paramètres la clé, le ton ou les deux, la valeur du 1er indice de TAB pour le caractère le plus à gauche et la valeur du 2ème indice de TAB correspondant à PV = 0. Ce sous-programme a pour nom CLYTON (I, J, K, L) où I indique la clé ou est à 0, J indique le ton ou est à 0, K et L étant les valeurs requises pour les 1er et 2ème indices de TAB.

L'algorithme est le suivant :
II (*) = I (*) ;
IF KE = I (1) THEN CALL CLYTON (CLF (1), TON (1), II - K (1), 0) ;
N = 2 ; LL = 1 ;
ICI : IF il n'y a pas d'éléments $\neq 0$ dans TAB (JJ, *, *) pour JJ \geq I (N)
THEN GOTO FIN ;
CALL CLYTON (CLF (N), TON (N), I (N) - K (N), - DK*LL) ;
IF II (N) + 100 - K (N) \neq II (N + 1)
THEN II (N) = II (N) + 100 - K (N) ; ELSE N = N + 1 ;
.../...

```
LL = LL + 1 ; GOTO ICICI ;  
FINI : ...
```

VII - 4 - d) TRADUCTION DE TAB EN L.S.

TAB va être traduit en une sous-chaîne de L.S. dont les codes des éléments seront rangés dans un tableau noté TABLS (1: nn). Les éléments terminaux du langage LS sont RC, BS, X (1), ..., X (99), C (1), ..., C (p-1) et C (p).

L'algorithme est le suivant :

```
L, LL = 1 ; J = plus grand JJ tel qu'il existe un élément ≠ 0 dans TAB (*,JJ,*) ;  
IF J < 0 THEN  
  Do JJJ = 1 To -J ;  
    TABLS (L) = 'RC' ; L = L + 1 ;  
  END ;  
LAHO : II = 0 ;  
Do IJ = I (1) - K (1) To I (1) - K (1) + 99 ;  
  IF TAB (II, J, 1) = 0 pour II ≥ IJ THEN GOTO IIICRJ ;  
  IF TAB (IJ, J, 1) = 0 THEN II = II + 1 ;  
  ELSE Do ;  
    TABLS (L) = 'X(II)' ; II = 0 ;  
    TABLS (L + 1) = 'C(TAB(IJ,J,1))' ; KK = 2 ; L = L + 2 ;  
    Do WHILE (TAB(IJ,J,KK) ≠ 0 ;  
      TABLS (L) = 'BS' ; TABLS (L + 1) = 'C (TAB (IJ, J, KK))' ;  
      L = L + 2 ; KK = KK + 1 ;  
    END ;  
  END ;  
END ;  
THEN  
TABLS (L) = 'RC' ; L = L + 1 ; IF J = 1 THEN L = L ; J = J - 1 ;  
IF il existe un élément ≠ 0 dans TAB (*,JJ,*) pour JJ ≤ J  
THEN GOTO LAHO ;  
IF J > 0 THEN Do ; Do JJJ = 1 To J ;  
  TABLS (L) = 'RC' , L = L + 1 ;  
  END ; LL = L ;  
END ; L = L - 1 ;
```

A la sortie de l'algorithme, L représente le nombre d'éléments de TABLS et LL est l'indice de TABLS correspondant au premier élément terminal de l'ensemble d'éléments terminaux de L.S. décrivant la ligne contenant TAB (0, 0, *).

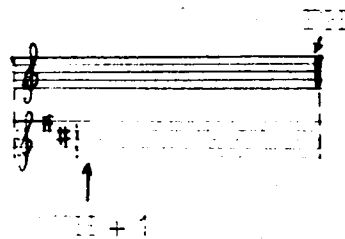
Remarque : les tests d'existence dans tous les algorithmes ci-dessus sont d'autant plus faciles que, si, pour TAB, on opte pour la représentation en mémoire proposée au § 1) a), seuls sont en mémoire les éléments non nuls.

VII - 4 - e) INSERTION DE TABLS DANS LA CHAÎNE L.S. DÉJÀ EXISTANTE.

Soit CHAINES (1 : pp) le tableau dans lequel est conservée la chaîne L.S. qui est mise à jour au fur et à mesure de l'élaboration des fonctions achevées.

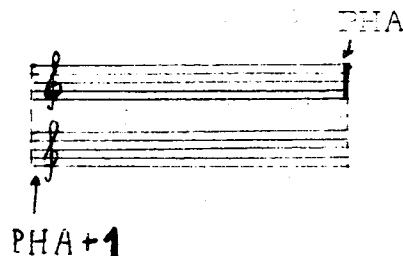
Le PH est une variable qui est incrémentée en ignorant les indications de clés et tons des débuts de portée :

ex. : partition à une voix



Parallèlement au PH est tenue à jour une variable, appelée PHA, qui diffère du PH dans le fait qu'elle est augmentée du nombre de caractères (en largeur) des indications de clés et tons des débuts de portée.

ex. : partition à une voix



Trois variables (IM, PHCH et NB), mises à jour en fin de traitement d'une fonction achevée, sont accessibles lors du traitement des fonctions achevées :

IM est l'indice dans CHAINLS correspondant à LL (pour TABLS), mais pour la fonction achevée précédente,

PHCH est le PHA correspondant à TAB (I (1) - K (1), 0, *)
mais pour la fonction achevée précédente,

NB est le nombre d'éléments de CHAINLS.

Posons PHLS = PHA correspondant à TAB (I (1) - K (1), 0, *)
pour la fonction en cours. L'algorithme est le suivant (on utilise un tableau intermédiaire INTTAB (1 : qq) pour ranger le résultat de la fusion de CHAINLS et TABLS)
/* Recherche, dans CHAINLS et TABLS, d'indices relatifs aux débuts des descriptions d'une même ligne */

IP = (PHLS - PHCH) / 100 * DK ; IF IP > 0

THEN Do ;

Do II = 1 To IP ; Do WHILE (CHAINLS (IM) ≠ 'RC') ;

IF IM > NB THEN Do ; CHAINLS (IM) = 'RC' ; GOTO SORT 1 ; END ;

IM = IM + 1 ; END ; SORT 1 : IM = IM + 1 ; END ; END ;

ELSE Do ;

Do II = 1 To - IP ; Do WHILE (TABLS (LL) ≠ 'RC') ;

IF LL > L THEN Do ; TABLS (LL) = 'RC' ; GOTO SORT 2 ; END ;

LL = LL + 1 ; END ; SORT 2 : LL = LL + 1 ; END ; END ;

/* Les lignes dont les premiers éléments sont CHAINLS (IM) et TABLS (LL) sont à superposer */

/* Début de fusion */ NB 1, NB2 = 1 ;

DECR : IM = IM - 1 ; LL = LL - 1 ;

Do WHILE ((CHAINLS (IM) ≠ 'RC') | (IM = 0)) ; IM = IM - 1 ; END ;

Do WHILE ((TABLS (LL) ≠ 'RC') | (LL = 0)) ; LL = LL - 1 ; END ;

IF IM + LL = 0 THEN GOTO FUSION ;

IF IM * LL ≠ 0 THEN GOTO DECR ;

IF LL = 0 THEN Do ;

Do IP = 1 To IM ; INTTAB (IP) = CHAINLS (IP) ;

END ; NBB = IM ; NB 1 = NBB + 1 ; END ;

ELSE Do ;

Do IP = 1 To LL ; INTTAB (IP) = TABLS (IP) ;

END ; NBB = LL ; NB2 = NBB + 1 ; END ;

FUSION : nous n'indiquerons, pour cette partie de l'algorithme, que la manière de procéder, la liste des instructions PL/I étant longue :

Il nous reste à fusionner CHAINLS (à partir de l'indice NB1) et TABLS (à partir de l'indice NB2) dans INTTAB (à partir de l'indice NBB) en examinant parallèlement CHAINLS et TABLS.

Dans les différents cas énumérés plus bas, CH1 représente indifféremment CHAINLS ou TABLS ; CH2 représente CHAINLS si CH1 est TABLS et inversement. Appelons I1 l'indice courant de CH1 et I2 celui de CH2. Pour simplifier les explications, on suppose que, dans les tableaux, tout élément X(n) (du langage L.S.) n'occupe pas un mais n éléments contenant chacun X(1). Soit N1 le nombre d'éléments de CH1 et N2 celui de CH2. Nous utiliserons le terme "introduire" pour indiquer en fait : "introduire dans INTTAB, à partir de l'indice courant, les p éléments précisés et augmenter de p cet indice". Nous examinons alors CH1 et CH2 et opérons en fonction des cas suivants (à tester dans l'ordre) :

CAS	OPERATIONS A EFFECTUER
$\left\{ \begin{array}{l} I1 > N1 \\ I2 > N2 \end{array} \right.$	Stop
$\left\{ \begin{array}{l} I1 > N1 \\ I2 < N2 \end{array} \right.$	Introduire CH2(I2), I2 = I2 + 1
$\left\{ \begin{array}{l} CH1(I1) = 'RC' \\ CH2(I2) \neq 'RC' \end{array} \right.$	Introduire CH2(I2), I2 = I2 + 1
$\left\{ \begin{array}{l} CH1(I1) = 'RC' \\ CH2(I2) = 'RC' \end{array} \right.$	Introduire 'RC', I1 = I1+1, I2=I2+1
$\left\{ \begin{array}{l} CH1(I1) = 'K(1)' \\ CH2(I2) \neq 'BS' \end{array} \right.$	Introduire CH2(I2), I1 = I1+1, I2 = I2 + 1
$\left\{ \begin{array}{l} CH1(I1) = 'C(n)' \\ CH2(I2) = 'C(m)' \end{array} \right.$	Introduire 'C(n)' 'BS' 'C(m)', I1=I1+1, I2=I2+1
CH1(I1) = 'BS'	Introduire 'BS' CH1(I1+1), I1 = I1+2

Après avoir attendu l'opération STOP, nous n'avons plus qu'à transférer INTTAB dans CHAINLS.

Remarque : on pourra compléter l'algorithme en rajoutant des instructions permettant de récupérer les valeurs à mettre dans L , $PHON$ et NB ; et $IPDA$ (voir chap. VIII - 2 - a - 2).

CHAPITRE VIII : DE L.S. A L'EDITION

VIII - § 1) DISPOSITIFS D'EDITION

VIII - 1 - a) GENERALITES

Les dispositifs d'édition venant à l'esprit au premier abord sont plutôt des dispositifs de photocomposition ou des tables traçantes, car les partitions musicales qui sont à deux dimensions et dont les éléments graphiques sont d'une très grande diversité quant à leur nombre et leurs multiples formes et tailles, font plutôt penser à des dessins qu'à des ensembles de caractères. Pour des questions de coût et même de vitesse d'édition, nous avons préféré essayer d'utiliser des organes de sortie classiques à caractères. Un terminal à clavier (du style I.B.M. 2740) ou une imprimante à pointes ne représente pas un investissement considérable, la préférence semblant aller actuellement à l'imprimante à pointes du fait de la possibilité offerte de définir arbitrairement les grilles de ses caractères de base ; et il n'est pas utopique d'envisager la fabrication d'une chaîne (contenant les caractères musicaux) pour une imprimante rapide ; les fichiers L.S. pourraient être enregistrés sur cassette et imprimés en off-line sur imprimante rapide comportant une chaîne spéciale. Le problème est de rendre les caractères contigus aussi bien horizontalement que verticalement.

VIII - 1 - b) LES CARACTERES

Un premier travail a été entrepris par des élèves du conservatoire de Grenoble : dresser, en regard d'un assez grand nombre de partitions diverses, une liste de toutes les formes possibles de représentation des notes, silences et symboles spéciaux.

A partir de là nous avons défini un ensemble de caractères de même taille dont l'assemblage permet de représenter des partitions musicales (voir annexe V). Nous nous sommes tout d'abord limités à 88 caractères nouveaux, de telle sorte qu'on puisse éventuellement fabriquer une boule spéciale pour 2740. Une partition pourrait ainsi être éditée en deux passages sur l'imprimante : le 1er pour imprimer les caractères spécifiquement musicaux au moyen de la boule spéciale, le 2ème pour imprimer les caractères classiques (en particulier pour les commentaires) au moyen d'une boule alphanumérique "standard". Bien entendu, les caractères de la boule spéciale sont superposables, pour permettre l'intersection de symboles musicaux (par exemple une liaison traversant une barre de note) et certains symboles ne peuvent même être obtenus qu'avec des superpositions ; pour ces superpositions on utilisera le caractère "backspace" de l'imprimante qui correspond d'ailleurs au symbole BS du langage L.S. Il va de soi que des superpositions de caractères provoquent un ralentissement à l'impression et nous avons donc défini les 88 caractères de telle sorte que des superpositions n'aient lieu, lors de l'impression, que dans des cas peu courants.

Pour la télévision, nous pouvons aussi définir un jeu de caractères affichables mais nous ne pouvons pas espérer les superposer ; en effet, l'affichage d'un caractère en un endroit de l'écran implique automatiquement l'effacement du caractère occupant éventuellement déjà l'emplacement. Nous avons donc dû trier toutes les superpositions possibles (ou, du moins, envisageables) en éliminant les cas très rares, ambigus ou non indispensables. Nous sommes arrivés à 416 caractères. Ce jeu peut d'ailleurs aussi être utilisé pour l'imprimante à pointes.

VIII - § 2) PROCEDURES D'EDITION

VIII - 2 - a) AFFICHAGE

VIII - 2 - a - 1 - Principes

Les caractères musicaux utilisés dans L.S. sont un sous - ensemble des 416 "caractères-télévision" ; c'est pourquoi nous avons introduit l'élément BS pour permettre, au moyen de superpositions, de représenter les caractères restants.

L.S. n'utilise qu'un sous-ensemble pour deux raisons essentielles :

- 1°) être compatible avec l'imprimante,
- 2°) faciliter la programmation (notamment pour le passage de L.I. (prétraité) à L.S.).

La méthode d'affichage d'un morceau de chaîne L.S. sur l'écran consiste à remplir une grille (correspondant à une "page-écran") avec les numéros des caractères apparaissant dans la chaîne ; cette grille sera alors transmise à la mémoire d'entretien . Les sous-chaînes de la forme :

C(m1) BS C(m2) BS C(m3)... BS C(mn)

sont traitées de la manière suivante : la configuration, résultant de la superposition de C(m1), C(m2), ..., C(mn) est calculée, puis la liste des 416 configurations des "caractères-télévision" est consultée ; si une configuration identique existe dans la liste le numéro de cette configuration est placé dans la grille ; si aucune configuration identique n'est trouvée, un numéro spécial sera placé dans la grille : soit, par exemple, zéro (c'est à dire pas de caractère) ou le numéro de la configuration correspondant au caractère *. (remarque : pour l'impression, si l'imprimante est une imprimante à boule, toute superposition est possible et il n'y aura donc, dans la partition, aucun caractère de numéro spécial).

VIII - 2 - a - 2 - Procédures de lancement

8 portées sont affichables, en même temps, sur l'écran. Nous pouvons convenir que la première portée est toujours : soit une portée relative à la voix en cours de création, soit une portée relative à une voix indiquée par l'utilisateur (par exemple lors de l'initialisation ou à tout moment à l'aide d'une touche de fonction supplémentaire). Soit IVC le numéro de la voix à laquelle correspond la première portée affichée, soit IV le numéro de la voix en cours de création et soit NV le nombre de voix. Le nombre (NBA) de portées affichées pour la voix en cours est déterminé par la séquence suivante :

```
IF IVC > IV THEN I = 0 ; ELSE I = 1 ; K = NV - (IVC - 1)
RETEST : IF K < 8 THEN IF K + NV < 8
THEN Do ; I = I + 1 ; K = K + NV ; GOTO RETEST ; END ;
ELSE IF K + IV <= 8 THEN I = I + 1 ;
NBA = I ;
```

Pour contrôler l'affichage, deux variables sont mises à jour à chaque lancement d'affichage ou modification de la chaîne L.S. :

IPTA pointe sur le premier élément (dans L.S.) correspondant au début de la "page-écran" affichée,

IPHA est le PHA correspondant à cet élément.

L'affichage aura lieu en fin de traitement d'une fonction achevée, donc le PHA courant est celui où devra débiter la fonction suivante. On procède alors comme suit :

Do WHILE (IPHA + NBA * 100 <= PHA) ;

IPHA = IPHA + 100 ;

IPTA = nouvelle valeur calculée en avançant (dans la chaîne L.S.) de DK lignes ;
END ;

Remplir la grille (à afficher) à partir de IPTA ;

Pour simplifier la programmation, il peut être convenu que l'affichage n'aura lieu qu'en fin de traitement des fonctions achevées pour lesquelles TAB a été traduit en L.S. et peut être effacé (c'est-à-dire les fonctions non suivies par X(n) (n<0) ou ne comprenant pas de débuts de liaisons ou de crescendos (ou décroscendos). Cependant, pour des questions de commodité d'utilisation, il est souhaitable que l'affichage soit lancé après chaque fonction achevée. Pour les fonctions ne provoquant pas l'effacement de TAB, il faudra donc remplir la grille à partir de la chaîne L.S. (qui ne contient pas encore la traduction de la fonction qui vient d'être traitée), puis compléter la grille directement à partir de TAB.

VIII - 2 - b) SORTIE SUR L'IMPRIMANTE

L'imprimante, seul son rôle d'éditeur étant considéré, n'est pas, comme la télévision, un instrument de contrôle mais un organe de sortie, en différé, des partitions. Pour la sortie sur l'imprimante, la commande PRINT est à utiliser (cf chap. III - 3 - a - 2).

Le programme d'écriture sur l'imprimante d'un fichier L.S. n'est lancé qu'une fois le fichier L.S. entièrement créé.

Le principe est très simple : on examine, un à un et en séquence, les éléments de la chaîne L.S., le chariot de l'imprimante étant initialement positionné en début de ligne :

- si on rencontre un élément $C(m)$, on lance l'écriture du m^{e} caractère,
- si on rencontre un élément $X(m)$, on lance l'écriture de m espaces,
- si on rencontre un élément BS, on lance l'écriture du caractère "backspace",
- si on rencontre un élément RC, on lance l'écriture des caractères "line feed" et "chariot return".

Remarque 1 : 2 passages sont nécessaires, le premier avec la boule spéciale, le second avec la boule alphanumérique "standard". Supposons que les caractères de la boule spéciale sont numérotés (dans L.S.) de 1 à 88 et ceux de la boule alphanumérique "standard" de 101 à 188. Lors du premier passage, les caractères dont le numéro est supérieur à 88 sont remplacés par des espaces ; lors du second passage, les numéros des caractères sont diminués de 100 et les caractères de numéro négatif sont remplacés par des espaces.

Remarque 2 : si l'imprimante est une imprimante à pointes, le jeu de caractères est le même que celui de la télévision ; un seul passage est nécessaire mais l'on procédera comme pour la télévision pour les combinaisons $C(m_1)$ BS $C(m_2)$... BS $C(m_n)$.



CHAPITRE IX : PROBLEMES PARTICULIERS

IX - § 1) MISE A JOUR (mode 4)

Pour la mise à jour des fichiers L.I. ou L.S. on disposera d'un pointeur, clignotant sur l'écran, dont le déplacement pourra être commandé au moyen de touches de fonctions.

IX - 1 - a) MISE A JOUR DANS LES CHAINES L.S. (mode 4 - 1)

Pas de problèmes particuliers ici : La position dans la chaîne L.S. correspond à celle du pointeur sur l'écran peut être aisément retrouvée, il sera donc possible de rajouter ou supprimer des caractères à partir de la position indiquée par le pointeur. Il faut bien noter, cependant, qu'il s'agit ici des caractères musicaux de base et donc que :

- 1^o) l'entreprise de modifications très substantielles risque d'être laborieuse,
- 2^o) les modifications ne seront pas transmises aux chaînes L.I. et, de ce fait, toute analyse sémantique ne pourra être faite que de l'ancienne partition.

La mise à jour des fichiers L.S. est donc réservée aux éditeurs pour leur permettre d'opérer quelques retouches (d'ordre plus particulièrement esthétique) sur des partitions presque définitives.

IX - 1 - b) MISE A JOUR DANS LES CHAINES L.I. (mode 4 - 2)

Pour des modifications importantes, il est nécessaire d'opérer sur les chaînes L.I. pour que, d'une part, les manipulations ne soient pas laborieuses et, d'autre part, une analyse sémantique de la nouvelle partition puisse être envisagée.

Plusieurs opérations sont prévues :

- supprimer les fonctions achevées pour lesquelles le PHA du pointeur-écran est compris entre le PHA de début des fonctions et le PHA de fin des fonctions ; ces fonctions peuvent être facilement retrouvées puisque le PHA de début des fonctions est indiqué dans les chaînes L.I. (cf chap. IV-2-f) et elles seront remplacées, dans la chaîne L.I. par une fonction X ;
- intercaler une fonction achevée entre deux fonctions achevées (la position de départ étant celle indiquée par le pointeur) ;
- supprimer ou rajouter une liaison rythmique (la première note étant indiquée par le pointeur) ;
- supprimer une liaison mélodique (la position de début étant indiquée par le pointeur) ;
- rajouter une liaison mélodique (les positions de début et fin étant indiquées par le pointeur).

Ce sont actuellement les seules opérations prévues. Elles posent peu de problèmes quant aux programmes de modification des chaînes L.I. à écrire, mais posent plus de problèmes quant au réaffichage immédiat du morceau de partition avec ses modifications :

ex. : CRO (NOT(40) LIA) DCR(~~NOT~~(40)) NOI (NOT(40))

La nouvelle chaîne résultant de la suppression de la fonction DCR est :

CRO (NOT(40) LIA) X (3) NOI (NOT(40))

Pour le réaffichage, on ne pourra pas se contenter d'effacer les caractères correspondant à la fonction DCR, il faudra aussi retraduire la liaison (reliant initialement la croche et la double croche) de telle sorte qu'elle relie la croche et la noire.

On pourra, par la suite, prévoir des opérations permettant, par exemple, le chevauchement de fonctions achevées, mais il faudra être très prudent, soit dans la définition de ces opérations, soit lors de l'utilisation qui en sera faite, car des contradictions, analogues à celles entraînées par le "retour-arrière" (chap. VII-1-b3), ou des ambiguïtés, entraînant la non-compatibilité avec DARMS (chap. IV - 4) risquent d'apparaître dans certains cas d'utilisation. Il n'est pas nécessaire d'exclure radicalement les opérations pouvant entraîner des contradictions ou ambiguïtés car, de toutes façons, certaines de ces opérations peuvent être utiles pour les éditeurs qui agiront en connaissance de causes et sont seulement intéressés par le résultat sur le papier.

IX - § 2) PROBLEMES DE RYTHME (sous le mode 3)

IX - 2 - a) PROBLEME D'APPROXIMATION

L'obligation d'indiquer, à l'initialisation du mode 3, les clé, ton et mesure initiaux ne suffit pas pour résoudre les problèmes de détermination des durées de notes jouées, le problème étant, en théorie, pourtant très simple (cf chap. V-1-b-2). Car, d'une part, il est difficile d'imposer à un interprète le respect rigoureux d'un certain tempo, surtout si l'interprète est un improvisateur laissant libre cours à son imagination et, d'autre part, quand bien même l'interprète se résoudrait à respecter le tempo, la précision de jeu ne sera pas celle d'une mécanique. Considérons, par exemple, le fait suivant :

Soit n la durée de base d'une noire ; la durée d'une ronde est alors de $4n$ et la durée d'une quadruple croche de $n/16$. Si l'on veut pouvoir différencier quatre quadruples croches d'un triolet de quadruples croches ou d'un quintolet de quadruples croches, il est évident que l'unité de base de l'horloge doit être bien inférieure à $n/16$ et dans bien des cas une durée supérieure à $n/(2 \times 16)$ sera répertoriée comme étant une durée valide et ne sera pas éliminée par les programmes d'approximation. Supposons qu'un interprète veuille jouer une note durant le temps de deux rondes reliées ; pour qu'à l'édition la deuxième ronde n'apparaisse pas reliée à une quadruple croche supplémentaire (ou à la première note d'un heptolet de quadruples croches) ou remplacée par une blanche cinq fois pointée, il faudra qu'il joue sa note avec une précision sur la durée de :

$$(n / (2 \times 16)) / (2 \times 4n) = 1/256 < 0,5 \%$$

IX - 2 - b) CHANGEMENT DE RYTHME-CHANGEMENT DE TON

L'ordinateur calcule les durées des notes et introduit les barres de mesures en fonction de la mesure donnée au départ. Mais si l'interprète commence un morceau à quatre temps et passe, en cours de jeu, à un rythme à trois temps, tout le morceau sera en fait édité à quatre temps. Pour un changement de ton le problème est le même: tout morceau sera édité avec, à la clé, les altérations indiquées au départ. La détermination de ces changements ne peut se faire qu'au moyen de puissants programmes d'analyse du morceau entier et l'édition sera donc nécessairement différée.

IX - 2 - c) UTILITE DU METRONOME

Mis à part le rôle de chef d'orchestre du métronome (qui impose le tempo à l'interprète), l'introduction dans la file d'entrée des tops de métronome (avec l'heure d'arrivée) peut permettre de supprimer le décalage, pouvant apparaître si la précision de jeu de l'interprète (ou même celle du métronome) n'est pas suffisante, au moyen de comparaisons entre l'heure de début des sons et l'heure des tops de métronome.

IX - 2 - d) REMARQUES

Ces problèmes d'approximation des durées des notes ont en fait été posés par souci de qualité de l'édition. Il est certain que, sans programmes d'approximation très puissants, des décalages des barres de mesure apparaîtront par suite du manque de précision temporelle des informations d'entrée ; les partitions obtenues seront peut-être inattendues et plus difficiles à lire, mais auront l'avantage d'être très fidèles au jeu de l'interprète.

Pour éviter les décalages sans nécessiter de programmes d'approximation monstrueux, une première approche peut être faite en concevant des programmes d'approximation (ou plutôt d'estimation) spécialisés dans certains styles de musique comme, par exemple, le jazz et plus particulièrement le piano "stride" : en effet, dans ce cas particulier caractérisé par un jeu de main gauche alternant entre une note grave et un accord plus aigu, de petits programmes d'analyse doivent suffire pour rechercher dans la file d'entrée les notes graves, ces notes débutant toujours sur un temps.

CHAPITRE X : PERSPECTIVES

Une partition ne donne jamais que les lignes directrices d'une composition musicale. Tout l'attrait d'un morceau de musique résulte en fait de l'harmonieuse conjonction de ces lignes directrices et de la personnalité d'un interprète. Une partition ne peut pas être le reflet total du génie d'un compositeur, car on y voit surtout l'aspect quantitatif de l'œuvre, l'aspect qualitatif étant bien incomplètement retracé à l'aide de commentaires et symboles spéciaux. La complexité des problèmes d'interprétation des signaux d'entrée découle de la non-universalité de l'écriture musicale habituelle : on voudrait en fait pouvoir retracer les lignes directrices d'une œuvre à partir du jeu d'un instrumentiste, ce qui implique nécessairement beaucoup d'ambiguïtés, donc des choix arbitraires. Il serait intéressant d'étudier la possibilité de définir un nouveau langage de description de la musique moins rigide et permettant de traduire, non seulement les sons, mais l'esprit dans lequel ils doivent être émis. D'autre part, dans l'écriture habituelle, les variations des sons sont toujours discrètes et les durées de base proportionnelles à un entier ou l'inverse d'un entier. La possibilité d'indiquer simplement des variations continues de sons et des durées absolument quelconques serait sûrement appréciée. Mais il ne faut pas pour autant abandonner la conception de programmes d'approximation pour le mode 3 car l'écriture musicale habituelle sera encore utilisée pendant longtemps. Certains musiciens peuvent, cependant, s'être défini une notation particulière de la musique qui ne diffère de la notation habituelle que par la forme des symboles et, dans ce cas, on peut leur laisser la possibilité de se définir eux-mêmes leur grille de caractères ; il va de soi qu'une notation fondamentalement différente de la notation habituelle obligerait à revoir la logique même des programmes d'édition.

Le but du projet est, bien entendu, de disposer d'un petit système compact directement utilisable par les musiciens mais les solutions proposées, en particulier pour les problèmes d'édition, laissent la porte ouverte à beaucoup d'applications.

On peut très bien envisager ce que pourraient être dans l'avenir les orchestres symphoniques : un ensemble de musiciens, chacun ayant devant lui un écran de télévision sur lequel défilerait la partition de l'orchestre ou seulement la partition qui l'intéresse ; le chef d'orchestre disposerait d'un clavier de fonction pour commander l'affichage : affichage à partir d'une certaine mesure, retour au début, etc. . . mais ce ne sont là que suggestions.

A N N E X E I

(Extraite d'un article de J. MERMET et Y. FANTINO [14])

- I - COMPOSANTS TECHNOLOGIQUES DU PERIPHERIQUE

a) Tube télévision :

C'est un matériel standard 625 lignes dont le déflecteur sera tourné de 90° afin de placer l'écran verticalement tout en conservant le balayage horizontal.

Le balayage sera réglable (comme dans les consoles alphanumériques) pour assurer une bonne géométrie des caractères.

b) * Générateur de caractères :

Il sera doté d'une mémoire morte standard contenant 7 circuits de 64 mots de 30 bits. Le signal vidéo sera fabriqué selon le même principe que dans les consoles alphanumériques à l'exception des lignes de portée qui se superposent à la mosaïque de caractères.

c) * Mémoire d'entretien :

L'image représentée sur l'écran correspondra à 96 lignes de 86 caractères chacune, chaque caractère ayant une largeur de 5 points et une hauteur de 6 points. La mémoire d'entretien contiendra donc 8 K mots de 9 bits plus un registre - tampon de 86 mots de 9 bits.

d) * Le clavier :

Il comportera 128 touches dont 61 touches de notes et 67 touches de fonctions (incluant les pédales). Les touches de notes proviennent de contacts placés sous un clavier d'orgue électronique simple. Les touches de fonctions proviennent soit d'un pédalier, soit des commutateurs de jeux de l'orgue.

e) * L'interface du clavier :

L'état du clavier est scruté périodiquement et tout changement se traduit par l'envoi du numéro de la touche manipulée avec une indication de temps, vers le calculateur.

La fréquence de scrutation du clavier sera telle qu'on puisse balayer au moins une fois le clavier par unité de temps.

L'unité de temps variera entre environ 0,5 et 4 ms suivant le réglage d'un métronome entre 0,5 et 2 z par l'utilisateur.

f) L'imprimante à pointes :

Elle sert à obtenir une copie sur papier de l'image contenue dans la mémoire d'entretien.

Une partie du générateur de caractères est réutilisé (mémoire morte).

La mécanique est disponible dans le commerce mais modifiée pour éliminer l'interligne.

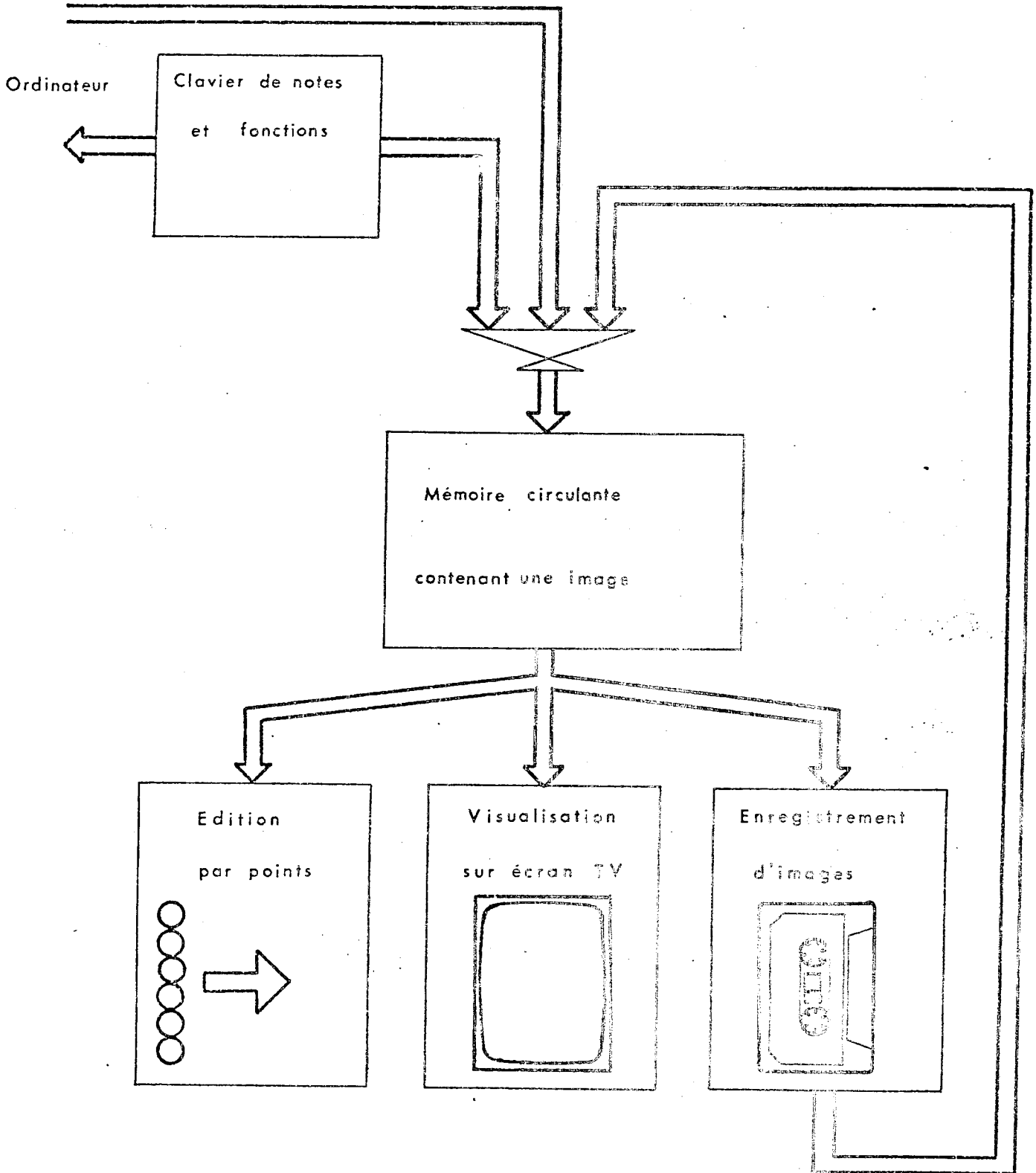
g) Clavier alphanumérique :

C'est le périphérique classique de dialogue avec l'ordinateur par conséquent notre système doit en posséder un, qui pourrait être, soit un terminal déjà existant dans le centre de calcul de l'installation, soit un clavier indépendant associé par exemple à l'imprimante à pointes.

h) L'enregistreur magnétique :

Le coût réduit des enregistreurs à cassettes permet d'envisager l'utilisation d'un tel appareil dans notre système, pour conserver des images de partitions et les reproduire à volonté.

PERIPHERIQUE MUSICAL : Eléments fonctionnels



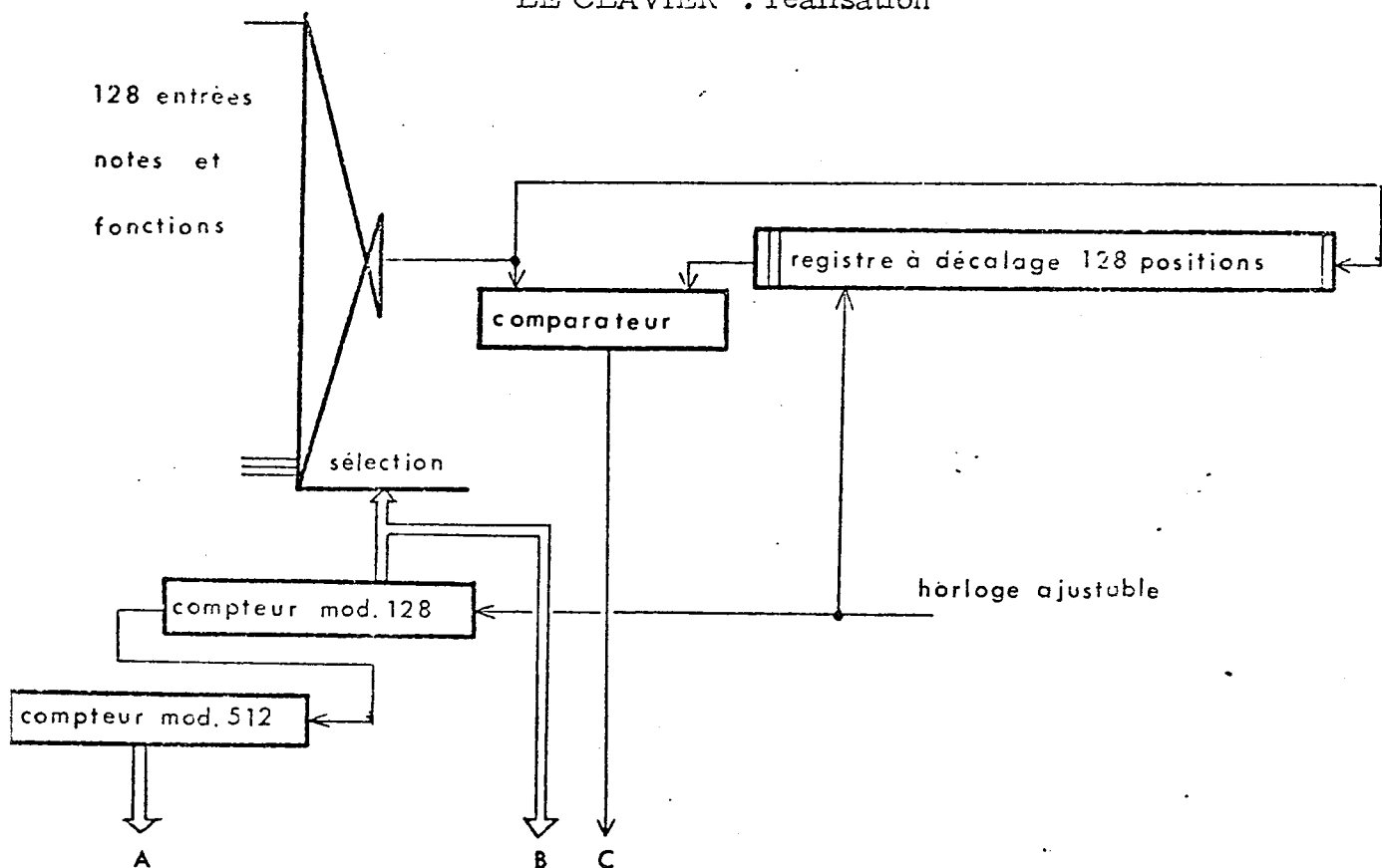
- II - PRINCIPE DE FONCTIONNEMENT

Le clavier d'orgue (5 octaves + 1) est le moyen d'entrée principal du périphérique. Il donne la hauteur des notes accompagnées de leurs instants de début et de fin. Ces signaux codés sont transmis à l'ordinateur où sont installés des programmes ayant pour but de leur associer une partition musicale. Les touches de "fonctions" sont utilisées pour envoyer des ordres au périphérique ou au calculateur ou des compléments d'information dans le cas où l'interprétation des signaux de notes bruts, serait trop complexe ou impossible.

L'image de partition élaborée par les programmes est envoyée vers la mémoire d'entretien du périphérique. La transmission complète d'une image à raison d'un caractère toutes les 200 micro-secondes demande environ 2 s. L'image circule en permanence dans la mémoire (registre MOS dynamiques) depuis laquelle des interfaces permettent de la visualiser (écran TV), de l'imprimer, de l'enregistrer.

Une image enregistrée pourra être visualisée et imprimée "off-line".

LE CLAVIER : réalisation



Fonctionnement :

L'horloge ajustable incrémente le compteur de sélection d'entrée et décale le registre de 128 bits d'une position. Si les entrées du comparateur sont différentes, une interruption (C) est générée vers l'ordinateur qui devra lire le numéro de la touche modifiée (B) et l'heure de la modification (A). La valeur de la touche (1 ou 0 = enfoncée ou relâchée) est dans tous les cas injectée à l'autre bout du registre pour la scrutation suivante. L'horloge sera réglable dans une plage telle que les débordements du compteur de temps fournissent des tops correspondant à la plage de fréquence classique d'un métronome ; ainsi dans tous les cas le clavier est scruté 512 fois entre chaque top de métronome.

- III - DESCRIPTION DETAILLEE DES ORGANES

1) La mémoire d'entretien d'image

a) L'image : l'image se compose d'une mosaïque de caractères adjacents, chacun comprenant 6 lignes de 5 points. Une image TV sera formée par deux trames entrelacées, l'une affichant les lignes paires des caractères et l'autre les lignes impaires.

Le jeu de caractères choisi comprend : 448 symboles, chaque caractère est codé sur 9 bits et circule en permanence dans la mémoire pouvant contenir 96 lignes de 86 caractères.

b) La mémoire : la mémoire se compose d'un registre à décalage de 86 mots de 9 bits contenant la ligne de caractères en cours d'affichage et d'un autre de 8192 mots de 9 bits contenant le reste de l'image. Avec le standard choisi une ligne de caractères sera utilisée pour 3 balayages consécutifs horizontaux de l'écran (lignes paires ou impaires). L'opération se passe en trois temps : 2 temps où le registre de ligne (86 pos) est recyclé sur lui-même, l'autre (8192 pos) n'étant pas modifié, un troisième temps où l'ensemble est décalé circulairement (dernier passage de la ligne en cours, renvoyée en bout d'image et remplacée par la ligne suivante).

c) Adressage de la mémoire : des compteurs de lignes et de colonnes sont gérés en synchronisme avec le décalage de la mémoire. Ils indiquent le caractère affiché par l'écran à chaque instant.

Deux registres d'adressage peuvent être chargés, incrémentés ou décrémentés. Un comparateur indique la coïncidence de la valeur de ces registres avec la valeur des compteurs lignes et colonnes, et autorise ainsi l'exécution d'une action sur la caractère en cours de visualisation, par exemple remplacement par un nouveau caractère.

d) Génération du signal vidéo : le caractère en cours adresse une mémoire morte de 448 mots de 30 bits contenant les symboles choisis. Les 30 bits fournis sont divisés en 6 lignes de 5 bits, un multiplexeur permet de choisir l'un de ces groupes de 5 bits (successivement lignes 0, 2, 4 puis à la trame suivante 1, 3, 5) qui est chargé dans un registre à décalage rapide dont la sortie constitue le signal à injecter à l'entrée de l'amplificateur vidéo.

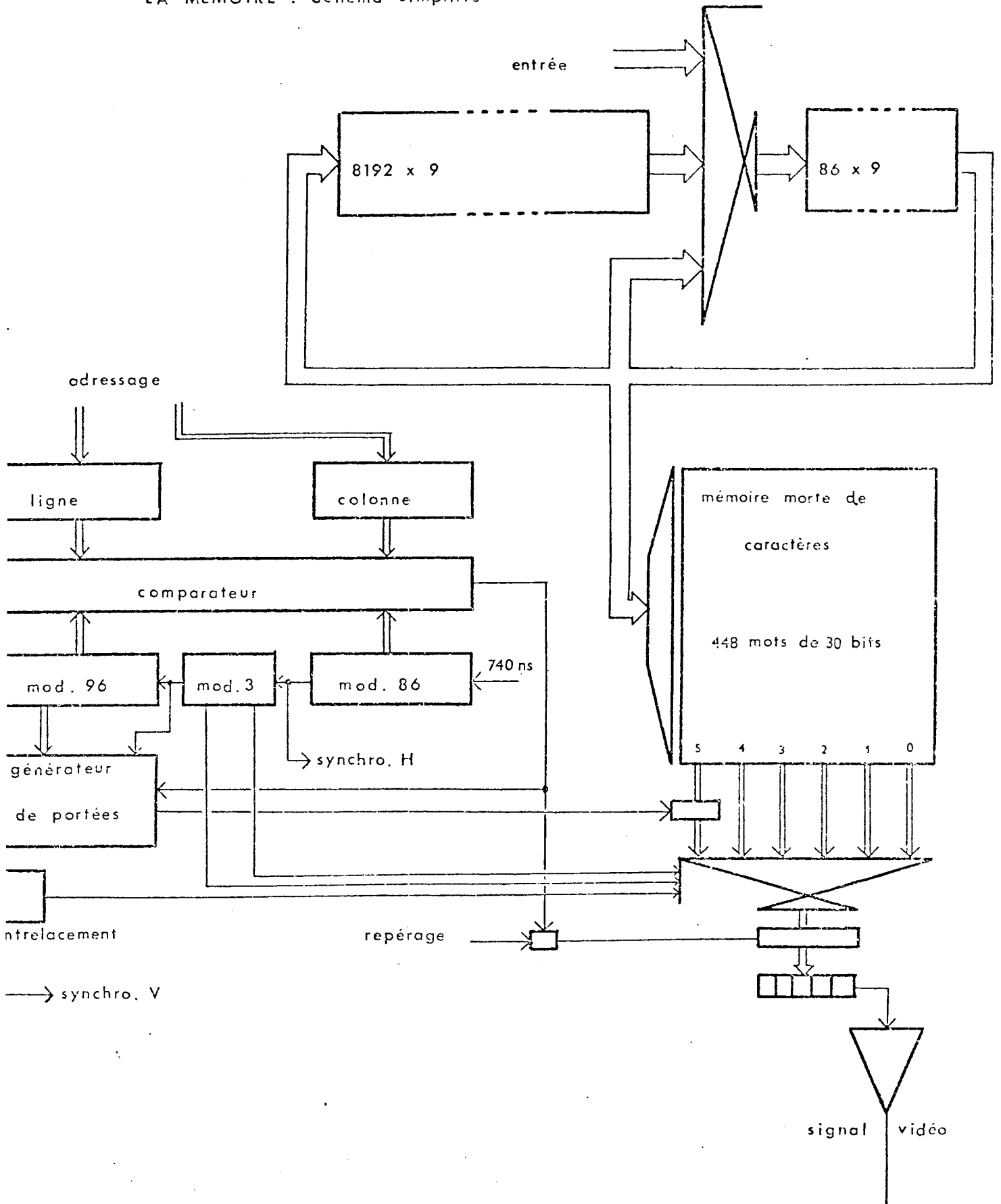
e) Dispositifs spéciaux :-Générateur de lignes de portées : on veut générer 8 portées. Une portée sera tracée en forçant la dernière ligne de points de cinq lignes de caractères successives. Six lignes de caractères séparent les portées. La première portée commence à la neuvième ligne de caractères en haut de l'écran.

Un compteur modulo 11 est maintenu à 0 pendant les 8 premières lignes de caractères, puis incrémenté à chaque ligne. Les combinaisons 1 à 5 de ce compteur provoquent le tracé des 5 lignes d'une portée.

- Repérage d'un caractère sur l'écran :

En vue de traitements spéciaux de l'image, ce dispositif permet de fournir l'adresse d'un caractère choisi sur l'écran par l'utilisateur. Ce caractère est pour l'utilisateur un caractère totalement blanc qui clignote. Il peut le déplacer à volonté (branchement d'une horloge sur les entrées ± 1 des registres d'adressage lignes et colonnes). L'adresse du caractère choisi peut être transmise vers l'ordinateur sur l'ordre d'une touche de fonction.

LA MEMOIRE : Schéma simplifié



ANNEXE II

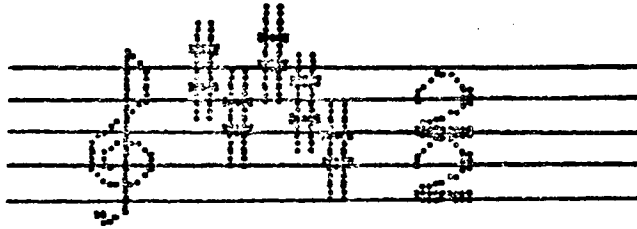
=====

RESULTATS D'EXEMPLES TRAITES PAR LES PROGRAMMES D'EDITION
(L'IMPRESSION A ETE REALISEE A L'AIDE D'UNE IMPRIMANTE
I.B.M. 2741 A BOULE "PLOTTER")

Handwritten musical notation on a five-line staff. The notation includes various note heads, stems, and beams. There are several measures with notes, some of which are grouped with beams. A vertical line is drawn across the staff, and there are some markings that appear to be a key signature or time signature, though they are difficult to read clearly. The notation is somewhat sparse and appears to be a sketch or a specific section of a larger piece.

Handwritten musical notation on a five-line staff. This staff contains more dense notation than the one on the left. It features many notes, some with stems and beams, and some with additional markings above them. There are several measures with notes, and a vertical line is drawn across the staff. The notation is more complex and appears to be a more developed section of a piece.

Les exemples suivants sont ceux donnés dans le chap.IV-2-g).
Les exemples 6,7 et 15 n'ont pas été traités, les liaisons, les
appoggiatures et les indications d'octave n'étant pas programmées.



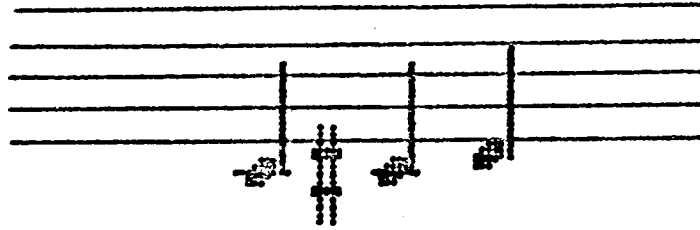
ex.1



ex.2



ex.3



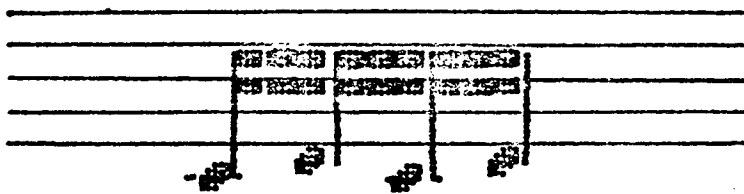
ex.4



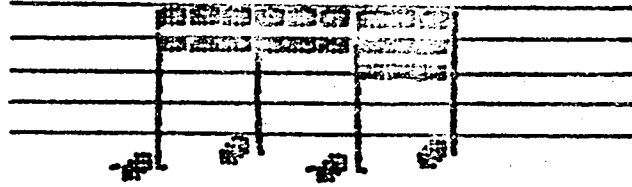
ex.5



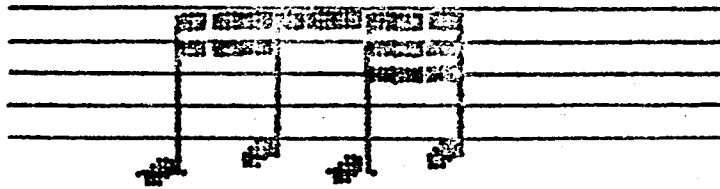
ex.8



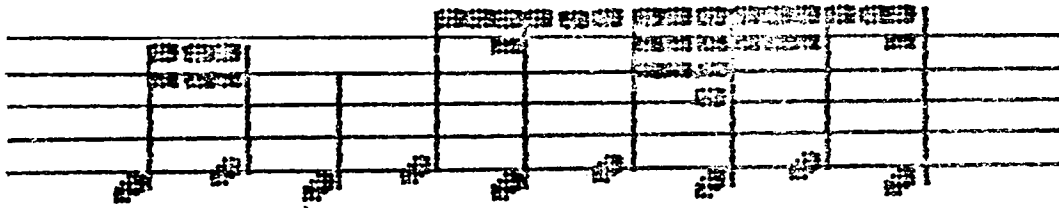
ex.9



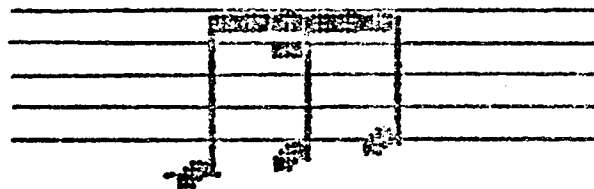
ex.10



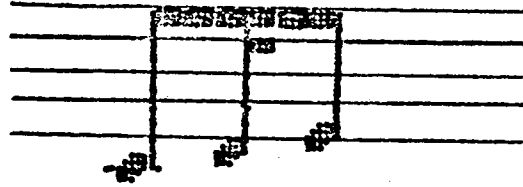
ex.11



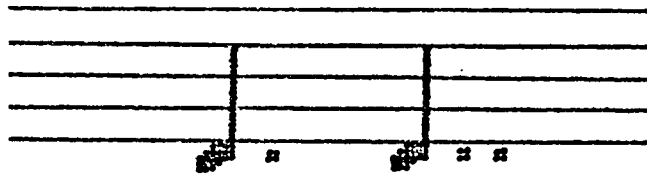
ex.12



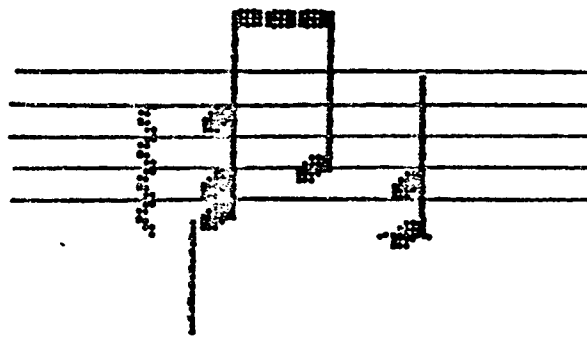
ex.13



ex.14



ex.16



ex.17

ANNEXE III

=====

LISTE DES CARACTERES POUVANT APPARAÎTRE DANS UNE CHAÎNE
DE CARACTERES DEFINIE PAR LA FONCTION GHA:

ABCDEFGHIJKLMN O PQRSTU VWXYZ

abcdefghijklmno pqrstuvwxyz

1 2 3 4 5 6 7 8 9 0

1 2 3 4 5 6 7 8 9 0














* + = - ? , . / x



ANNEXE IV

=====

TOUCHES DE FONCTIONS UTILISEES SOUS LE MODE 2 :

N°	marque de repérage	code mnémotechnique	Information complémentaire à entrer au clavier alphanumérique	Signification
1	↑	PRH		→ vers le haut
2	→	PRD		déplacement d'un espace du pointeur } <ul style="list-style-type: none"> ↘ à droite ↙ vers le bas ← à gauche
3	↓	PRB		
4	←	PRG		
5		SIL	un code	
6		NDG	un code	indique un doigté, spécifié par le code, à éditer à l'emplacement des PH et PV courants
7		NLT	un code	indication de n-olet (précisé par le code) concernant les notes et silences suivants (jusqu'à l'enfoncement de la touche 8)
8		FNT		fin n-olet
9		BEH		orientation des barres des notes vers le haut
10		BEB		orientation des barres de notes vers le bas
11		APP		les notes suivantes seront des oppoggiatures
12		HPT		les notes et silences précédents doivent être pointés
13		LIA		liaison rythmique
14		CLE	un code	pour indiquer un changement de clé
15		TON	un code	pour indiquer un changement de ton
16		MES	un code	pour indiquer un changement de mesure
17		BAM		pour indiquer une barre de mesure

18	♀	T18
19	♯	T19
20	◡	T20
21	⚡	T21
22	⚡	T22
23	∞	T23
24	∞	T24
25	∧	T25
26	∨	T26
27	·	T27
28	#	ADI
29	⊗	ADD
30	b	ABE
31	bb	ADB
32	b	ABC
33	o	VRO
34	o	VBL
35	♪	VNO
36	♪	VCR
37	♪	VDC
38	♪	VTC
39	♪	VQC
40	∧	DBC
41	∨	FNC
42	D)	DLM
43	F)	FLM
44	△	T30
45	⊕	T31
46	⊗	T32
47	◇	PAC
48	X	PAX
49	gta	PAN
50	'...'	CHA
51	ANN	ANN

chaîne de caractères

symbole ♀
 symbole ♯
 symbole ◡
 symbole ⚡
 symbole ⚡
 symbole ∞
 symbole ∞
 symbole ∧
 symbole ∨
 pizzicato

le symbole
 doit être édité
 à l'emplacement
 des PH et PV
 courants

pour imposer une
 altération différente de
 l'altération logique

symbole o
 symbole ♪
 symbole ♪
 symbole ♪
 symbole ♪
 symbole ♪

le symbole doit
 être édité
 à l'emplacement
 des PH et PV
 courants

début crescendo
 fin crescendo
 début liaison mélodique
 fin liaison mélodique

à éditer
 suivant les
 PV et PH
 courants

symbole △
 symbole ⊕
 symbole ⊗

le symbole doit
 être édité à
 l'emplacement
 des PH et PV
 courants

les notes suivantes seront des notes carrées

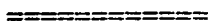
les notes suivantes seront des notes en croix

les notes suivantes ne doivent pas être mises à l'octave

pour éditer la chaîne de caractères donnée à partir de l'emplacement des PH et PV courants

pour annuler la fonction en cours

A N N E X E V



ESTHETIQUE DES CARACTERES MUSICAUX :

La composition musicale ou l'analyse de style à l'aide d'un ordinateur intéresse surtout les musiciens scientifiques. Par contre, un terminal, tel qu'il est présenté dans le présent ouvrage, est destiné à tout musicien même amateur que les problèmes internes d'informatique n'intéressent en général pas. Un effort particulier doit donc être fait dans un souci de qualité de présentation des résultats et, en particulier, de qualité de l'édition du point de vue de l'esthétique ; car un musicien profane sur le plan informatique admettra difficilement des partitions dont le graphisme est sensiblement différent de celui auquel il est habitué même si ces différences sont prétexte à simplification des programmes ou réduction du nombre de caractères musicaux internes.

Le choix des caractères musicaux a donc été fait de telle sorte que les assemblages, correspondant à des éléments de partition (notes, groupes de notes, silences, clés, etc...), se rapprochent le plus possible de la notation habituelle. Bien sûr, nous avons réussi à nous limiter à 88 caractères pour pouvoir éventuellement utiliser une imprimante à boule, pourquoi ne pas alors essayer de nous limiter à moins ? Si l'on constate qu'une diminution du nombre des caractères entraîne une augmentation du nombre de superpositions nécessaires pour obtenir toutes les représentations souhaitées et sachant que qui dit "superposition" dit "backspace" sur l'imprimante, on comprendra qu'une réduction supplémentaire du nombre de caractères de base serait nuisible à la vitesse d'impression des partitions. Les 416 "caractères-télévision" ont été obtenus en rajoutant aux 88 "caractères-imprimante" les caractères résultant des superpositions nécessaires pour l'obtention des représentations musicales souhaitées. Une diminution du nombre des "caractères-télévision" ne serait possible qu'en supprimant un certain nombre de superpositions (d'où une perte d'efficacité : par exemple une liaison ne pourrait plus traverser une barre de note) ou en modelant différemment les caractères de base (d'où un éloignement de la notation habituelle.

TWILIGHT

Ernest Gagne

Slowly

7

bass sustained

P

Ped.

*

Photo en réduction d'un exemple construit
point par point et édité sur imprimante rapide
pour vérification de la qualité de l'édition.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

0 1 2 3 4 5 6 7 8 9

! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { | } ~

! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { | } ~

Photo en réduction de différents assemblages de caractères de base édités sur imprimante rapide.

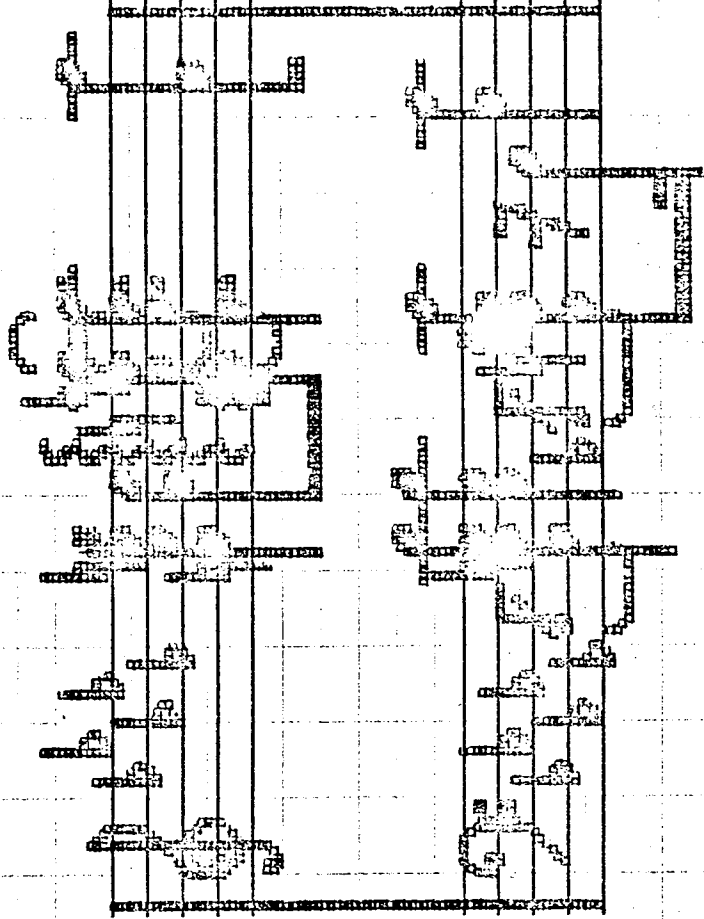


IMAGE DISPLAY ECHELLE : 10/10

Essai d'édition sur traçeur Benson
à partir d'une image composée sur l'écran
d'un poste d'affichage I.B.M. 2250.

ANNEXE VI

=====

LISTE DES 416 CARACTERES
DEFINIS POUR LA TELEVISION.

					*
					*

			*		
		*			
	*				

		*	*		
				*	*

		*		*	
	*		*		
				*	

*					
	*				
			*		
				*	

*					
*					
		*			
	*		*		
				*	*

			*	*	*

			*		
			*		
				*	
				*	
					*

		*	*		
	*	*			
		*			
		*			
			*		
				*	

*	*	*			

				*	
			*		
			*		
				*	
				*	
					*

*				*	*
*	*		*	*	*
		*	*	*	*
		*	*	*	*
		*	*	*	*
		*	*	*	*

PAGE 013

CAMBRIDGE MONITOR SYSTEM

FILE: CACTCS DATA

*	*	*	*	*
				*
				*
				*
				*
*				

				*
			*	
			*	
			*	
		*	*	
*				

*	*			
		*		
		*		
		*		
		*		
		*		

		*		
		*		
		*		
		*		
		*		
		*		

*	*	*	*	*
*	*	*	*	*

		*	*	*
	*	*	*	*
			*	*
				*
			*	*
	*	*	*	*
				*
			*	*
				*
			*	*

*				
	*			
		*		
	*			
	*			
	*			
*	*	*	*	*

*				
		*		
	*			
	*			
	*			
	*			
*	*	*	*	*

*	*	*	*	*
*	*	*	*	*

		*	*	*
	*	*	*	*
			*	*
				*
			*	*
	*	*	*	*
				*
			*	*
				*
			*	*

*				
	*			
		*		
	*			
	*			
	*			
*	*	*	*	*

		*		
	*			
	*			
	*			
	*			
*	*	*	*	*

PAGE 018

CAMBRIDGE MONITOR SYSTEM

DATE: 1981-03-10

					*		*
						*	
					*		
						*	

							*

					*		
					*		
					*		
					*		
					*		
					*		

					*		
					*		
					*		
					*		
					*		
					*		

					*		
					*		
					*		
					*		
					*		
					*		
					*		
					*		*

					*		
					*		
					*		
					*		
					*		
					*		
					*		
					*		*

					*		
					*		
					*		
					*		
					*		
					*		
					*		
					*		*

					*		
					*		
					*		
					*		
					*		*

					*		*
					*		
					*		
					*		
					*		
					*		
					*		
					*		
					*		*

							*
					*		
					*		*
					*		*
					*		*

					*		
					*		
					*		*
					*		*
					*		*
					*		*
					*		*
					*		*
					*		*

					*		
					*		
					*		
					*		
					*		*

		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*

		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*

		*	*
*	*	*	*

		*	*
		*	*
		*	*
		*	*
*	*	*	*

*	*	*	*
*	*	*	*
*	*	*	*

		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
		*	*
*	*	*	*

*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*

*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*

*	*	*	*
*	*	*	*

*	*	*	*
*	*	*	*

*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*

*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*

BIBLIOGRAPHIE

- [1] J.K. RANDALL
"A report from Princeton"
Perspectives of New Music vol. 3 n° 2 Spring-Summer 1965
- [2] James C. TENNEY
"Sound generation by means of a digital computer"
Journal of Music Theory vol. 7 n° 2 Spring 1963
- [3] Barry S. BROOK
"Music Bibliography and the Computer"
In : Papers from the West Virginia University Conference on Computer Applications in Music West Virginia University Library Morgantown 1967
- [4] E. A. BOWLES
"Computerised research in the humanities ; a survey"
ACLS Newsletter, Special Supplement June 1966 p. 38
- [5] A. FORTE
"A program for the analytic reading of scores"
Journal of Music Theory vol. 10 n° 2 Winter 1965
- [6] M. BABBIT
"The use of computers in musicological research"
Perspectives of New Music vol. 3 n° 2 Spring-Summer 1965
- [7] J. ROTHGEB
"Some uses of mathematical concepts in theories of Music"
Journal of Music Theory vol. 10 n° 2 Winter 1965

- [8] M. KASSLER
"A sketch of the use of formalized languages for the assertion of Music"
Perspectives of New Music vol. 1 n° 2 Spring-summer 1963
- [9] Raymond F. ERICKSON
"The uses of computers in Music : A state of the Art report"
I.B.M. Systems Research Institute, New York City 1970-71
- [10] H.B. LINCOLN
"Toward a computer typography for music research : a progress report"
In : papers from the IFIP congress 1971 (Ljubljana, August 1971)
- [11] Barry S. BROOK, editor
"Symposium II : Input languages to represent Music"
Musicology and the computer : Three Symposia, The City University
of New York Press, New York 1970
- [12] D. JAEGER
"Entrée accélérée avec édition instantanée de partitions musicales à
l'aide d'un calculateur"
Séminaire de l'I.M.A.Mu, Paris, Mars 1972
- [13] D. JAEGER
"Un périphérique d'ordinateur à l'usage des musiciens : programmes
d'utilisation"
Textes des conférences des journées d'informatique musicale, C.N.R.S.
Paris, Octobre 1973
- [14] J. MERMET, Y. FANTINO
"Un périphérique d'ordinateur à l'usage des musiciens : composants tech-
nologiques du périphérique"
Textes des conférences des journées d'informatique musicale, C.N.R.S.
Paris, Octobre 1973

TABLE DES MATIERES

	page
CHAPITRE I : INTRODUCTION	7
I - § 1) Généralités	7
I - § 2) Mode de fonctionnement du système proposé	8
CHAPITRE II : DESCRIPTION GENERALE DU PERIPHERIQUE	11
II - § 1) Configuration générale	11
II - 1 - a) Description de la configuration	12
II - 1 - b) Clavier de piano	13
II - 1 - c) Clavier de fonctions	13
II - 1 - d) Pédalier	13
II - 1 - e) Métronome électrique	13
II - 1 - f) Ecran de télévision	13
II - 1 - g) Clavier alphanumérique	13
II - 1 - h) Imprimante	14
II - 1 - i) Autres dispositifs	14
II - § 2) Principes généraux de fonctionnement	14
CHAPITRE III : LE LANGAGE DE COMMANDES	15
III - § 1) Principes généraux	15
III - § 2) Notion de fichiers	16
III - 2 - a) Fichiers L.I.	16
III - 2 - b) Fichiers L.S.	17
III - § 3) Les modes	18
III - 3 - a) Mode 1	18
III - 3 - a - 1 - Généralités	18
III - 3 - a - 2 - Commande PRINT	19
III - 3 - a - 3 - Commande EDITION	19
III - 3 - a - 4 - Commande PIANO	20
III - 3 - b) Mode 2	21
III - 3 - c) Mode 3	21
III - 3 - d) Mode 4	22

CHAPITRE IV : LE LANGAGE INTERNE (L.I.)	23
IV - § 1) Généralités	23
IV - 1 - a) Pourquoi L.I. ?	23
IV - 1 - b) Principales indications d'esthétique	24
IV - 1 - c) L.I. : langage de fonctions	24
IV - § 2) Définition formelle	25
IV - 2 - a) Groupe I	26
IV - 2 - b) Groupe II	27
IV - 2 - c) Groupe III	28
IV - 2 - d) Groupe IV	29
IV - 2 - e) Groupe V	30
IV - 2 - f) PH et PV	30
IV - 2 - g) Exemples	31
IV - § 3) Représentation en mémoire	34
IV - § 4) Compatibilité avec DARMS	36
CHAPITRE V : PASSAGE DES SIGNAUX D'ENTREE A L.I. ET A L.I. PRETRAITE	39
V - § 1) Correspondance entre les informations d'entrée et L.I.	39
V - 1 - a) Mode 2	39
V - 1 - b) Mode 3	41
V - 1 - b - 1 - file d'entrée	41
V - 1 - b - 2 - passage à L.I.	42
V - § 2) Passage de L.I. à L.I. prétraité	42
V - 2 - a) Traitement sur les notes et altérations	43
V - 2 - a - 1 - Généralités	43
V - 2 - a - 2 - Algorithmes	44
V - 2 - a - 3 - Problème du "retour-arrière"	49
V - 2 - b) Restrictions différenciant L.I. prétraité de L.I.	50
V - 2 - c) Différence entre le mode 2 et le mode 3	51
CHAPITRE VI : LE LANGAGE DE SORTIE (L.S.)	53
VI - § 1) Caractères musicaux	53
VI - § 2) L.S.	53

CHAPITRE VII : DE L.I. PRETRAITE A L.S.	55
VII - § 1) Principes généraux	55
VII - 1 - a) Tableau transitoire	55
VII - 1 - b) Principes de traitement	56
VII - 1 - b - 1 - Généralités	56
VII - 1 - b - 2 - Problèmes de LIA, DBC et FNC, DLM et FLM	56
VII - 1 - b - 3 - Problèmes de la fonction X	58
VII - 1 - b - 4 - Tests de passage à la fonction suivante	58
VII - § 2) Organisation du traitement d'une fonction achevée	59
VII - 2 - a) 1er pas : examen de la fonction ; traitement de tout ce qui n'est pas relatif à une note	59
VII - 2 - b) 2ème pas : traitement des appoggiatures	60
VII - 2 - c) 3ème pas : traduction du contenu de ITFONC relatif aux notes non appoggiatures	61
VII - 2 - c - 1 - BEH ou BEB	61
VII - 2 - c - 2 - Principe d'examen de la chaîne	62
VII - 2 - c - 3 - CHBARNOT	62
VII - 2 - c - 4 - Sous-programme ACCORD	63
VII - 2 - d) 4ème pas : n-olets, indicateurs d'octaves, crescendos et liaisons	64
VII - § 3) Algorithmes particuliers	64
VII - 3 - a) Elaboration des crescendos	64
VII - 3 - b) Elaboration des barres des notes	66
VII - 3 - b - 1 - Calcul de la disposition des barres horizontales communes	67
VII - 3 - b - 2 - Recherche de la place disponible dans TAB	68
VII - 3 - c) Elaboration des accords	70
VII - 3 - c - 1 - Forme des notes	70
VII - 3 - c - 2 - Position des notes et de la barre de note	72
VII - 3 - c - 3 - Placement des notes	73
VII - § 4) De TAB à L.S.	73
VII - 4 - a) Généralités	73
VII - 4 - b) Compléments sur le passage de L.I. à TAB pour les fonctions CLE et TON	75

VII - 4 - c) Remaniement de TAB	77
VII - 4 - c - 1 - Réorganisation du contenu	77
VII - 4 - c - 2 - Insertion des clés et tons des débuts de ligne de partition	77
VII - 4 - d) Traduction de TAB en L.S.	78
VII - 4 - e) Insertion de TABLS dans la chaîne L.S. déjà existante	79
 CHAPITRE VIII : DE L.S. A L'EDITION	 83
VIII - § 1) Dispositifs d'édition	83
VIII - 1 - a) Généralités	83
VIII - 1 - b) les caractères	83
VIII - § 2) Procédures d'édition	84
VIII - 2 - a) Affichage	84
VIII - 2 - a - 1 - Principes	84
VIII - 2 - a - 2 - Procédures de lancement	85
VIII - 2 - b) Sortie sur l'imprimante	86
 CHAPITRE IX : PROBLEMES PARTICULIERS	 89
IX - § 1) Mise à jour (mode 4)	89
IX - 1 - a) Mise à jour dans les chaînes L.S. (mode 4-1)	89
IX - 1 - b) Mise à jour dans les chaînes L.I. (mode 4-2)	89
IX - § 2) Problèmes de rythme (sous le mode 3)	91
IX - 2 - a) Problème d'approximation	91
IX - 2 - b) Changement de rythme - changement de ton	91
IX - 2 - c) Utilité du métronome	92
IX - 2 - d) Remarques	92
 CHAPITRE X : PERSPECTIVES	 93
ANNEXE I : COMPOSANTS TECHNOLOGIQUES DU PERIPHERIQUE	95
ANNEXE II : RESULTATS D'EXEMPLES TRAITES PAR LES PROGRAMMES D'EDITION	103
ANNEXE III : LISTE DES CARACTERES POUVANT APPARAÎTRE DANS UNE CHAÎNE DE CARACTERES DÉFINIE PAR LA FONCTION CHA	109
ANNEXE IV : TOUCHES DE FONCTIONS UTILISÉES SOUS LE MODE 2	111
ANNEXE V : ESTHÉTIQUE DES CARACTÈRES MUSICAUX	113
ANNEXE VI : LISTE DES 416 CARACTÈRES DÉFINIS POUR LA TÉLÉVISION	117
BIBLIOGRAPHIE	153