SINTETIZADOR MS-80: PROTÓTIPOS DE HARDWARE E SOFTWARE

Celso Aguiar*

INTRODUÇÃO

Fugindo um pouco a horizontalidade da visão oferecida nos cursos de graduação em Composição, foi criado em 1984 junto ao Departamento de Engenharia Elétrica da Escola Politécnica da Universidade Federal da Bahia o projeto de pesquisa "SSCD" (Síntese Sonora com Computadores Digitais) como uma iniciativa de aluno (Celso Aguiar – Composição e Regência/Escola de Música da UFBa.) na tomada de primeiros contactos com tecnologias na área de síntese com computadores visando a pesquisa e produção de Música Eletrônica.

Apesar dos investimentos realizados no país ultimamente em ciências da informação aplicada a todos os setores do conhecimento, temos a impressão de que pouco tem sido realizado na aquisição da tecnologia de uso ou desenvolvimento de ferramentas para a síntese ou composição com computadores. Desconhecemos, apenas como exemplo, algum sucesso na implementação de programas modelo MUSIC N (Loy, 1985) em qualquer instituição superior de ensino ligada à Música no país. Tais iniciativas deverão ser encampadas, com certeza, por profissionais ou não em música com alguma vontade ou aptidão para as ciências da computação e eletrônica digital.

Considerando que os desafios deveriam ser enfrentados por inteiro, chegamos a formação de uma equipe em sistema de coorientação envolvendo os Departamentos de Engenharia Elétrica onde foi encontrado estímulo e respaldo ao estabelecimento do projeto (Prof. Raimundo R. Cavalcante, Ph.D em Sistemas Eletrônicos, Cranfield – Inglaterra) e Música (Prof. Jamary Oliveira, Ph.D em Composição, Austin – EUA) buscando satisfazer a natureza interdisciplinar da pesquisa.

Com recursos obtidos junto ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), foi adquirido um microcomputador e componentes para a montagem de uma interface digital para geração de som através de conversores D/A de 8 bits que vem operando satisfatoriamente.

Com liberação de recursos em agosto de 87 para uma nova etapa do projeto, demos início à compra de equipamentos para montagem

*Graduando em Composição – Escola de Música da UFBa.

de um razoavelmente complexo sistema de síntese envolvendo unidades de geração de som com arquitetura calcada no protótipo anteriormente mencionado e contrôle por processamento distribuído.

HARDWARE

Protótipo e Filosofia do Sistema

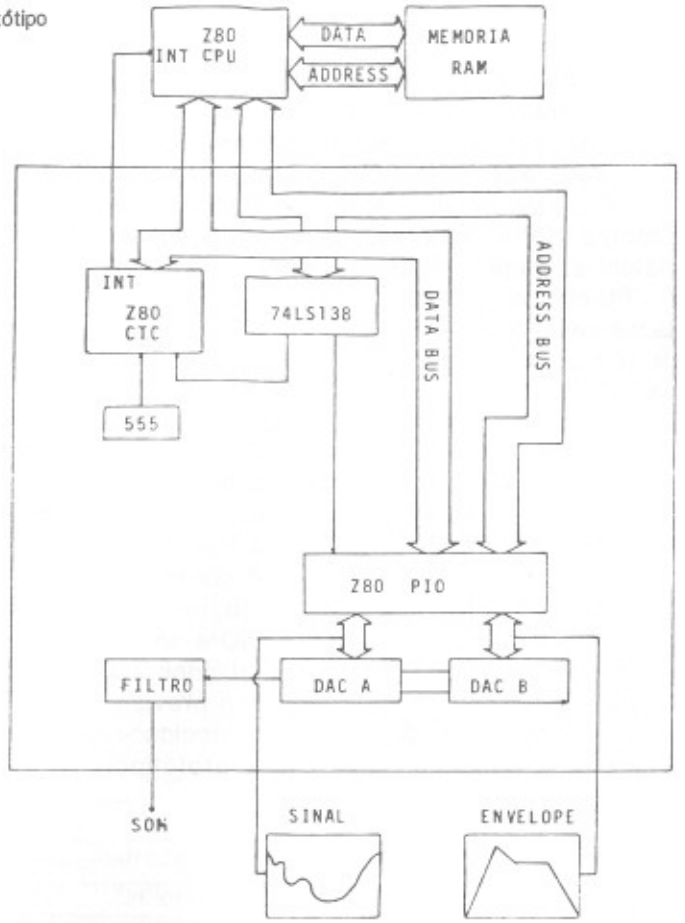
Na aplicação de computadores à síntese sonora, aqui descrita, optamos pela utilização de microprocessadores dedicados deixando de lado o uso de grandes sistemas onde o músico é apenas um usuário de uma cadeia operando em "time-sharing". Aos preços atuais dos microprocessadores as vantagens são muitas, dentre elas a estimulante perspectiva do projeto de sistemas operando em tempo-real como forma de encurtar o caminho em direção à experiência com o som (Alles, 1980). No çurso da pesquisa nos definimos, portanto, por um sistema de microcomputador interligado a placa-interface com conversores D/A de 8 bits e dedicado à síntese em tempo-real, através do método de consulta a tabelas na memória ("table look-up") (Moore, 1985).

O diagrama do Protótipo (Fig. 1) procura destacar os fatores mais importantes desenvolvidos em hardware. Além do protótipo em si enfatizamos o papel desempenhado pela CPU e memória do sistema onde são armazenados programas e blocos de tabelas definindo o timbre a ser escutado.

A CPU comunica-se com o protótipo, através, basicamente, dos barramentos de endereço e dados. Estes chegam aos conversores por meio de porta paralela de comunicação. O Z80-PIO encarrega-se, portanto, de lançar dados de amplitude de sinal ou envelope aos devidos conversores. A função do Z80-CTC, por outro lado, é a de definir o ritmo dessas amostragens por meio de interrupções da CPU (pinos INT do CTC e CPU). Sua incorporação ao sistema implica diretamente numa simplificação da programação (subrotinas de interrupção) e liberação da CPU para outras tarefas (ex: varredura de teclado quando de execução tempo-real).

Dispondo de dois conversores digitais-analógicos pudemos chegar a um sistema em "cascata" onde um dos conversores modula a amplitude do sinal sonoro emitido por outro conversor (Fig. 1).

Fig. 1: Diagrama Protótipo



Uma vantagem que pode ser deduzida do que acima foi descrito é a de que quaisquer formas de envelope de amplitude podem ser impostas ao sinal sonoro através do DACB, bem como quaisquer formas de onda sonora podem ser emitidas pelo módulo oscilador (DACA). São possíveis, portanto, simulações de uma grande variedade de instrumentos musicais, através de qualquer técnica de síntese que tenha sido implementada no sistema para a sua criação através de programação. Em particular, instrumentos com variação dinâmica do timbre são obtidos com facilidade através da modificação gradativa da forma de onda lançada ao conversor A,durante a execução de uma nota.

MS-80: Sistema em Processamento Distribuído

O sistema de processamento distribuído, que ora encontra-se em início de montagem, situa-se como uma alternativa possível na integração de vários canais sonoros independentes em um sintetizador complexo com tecnologia de multi-processamento.

Projetamos um sistema constituído de CPU mestre e vários processadores (CPUs escravos) dedicados apenas a tarefa de síntese, com sistemas independentes de memória e conversão de amostras portanto. A CPU central atua como gerenciadora (ou "regente") das tarefas executadas pelos escravos. Este gerenciamento se dá através de mecanismos de comunicação entre mestre e escravos sendo possíveis os dois sentidos para solicitação de dados.

O funcionamento de um escravo consiste em receber arquivos gerados e transmitidos pelo mestre com vistas a execução de tarefas semelhantes às citadas para o desenvolvimento do protótipo. Com este setor do hardware já desenvolvido e testado resta-nos o projeto do sistema que integre escravos permitindo a transferência de arquivos.

Uma placa escrava (Fig. 2) é constituída de CPU Z80-B, memória EPROM, RAM dinâmica com 64 Kbytes, duas portas de comunicação com o mestre, dois CTC's, uma PIO e dois DAC's. Frisamos que todos os dispositivos da família Z80 utilizados estarão operando segundo a versão Z80-B com clock do sistema previsto para 6 MHz, podendo-se prever portanto uma duplicação da velocidade de amostragem em relação à faixa que vimos operando com o protótipo.

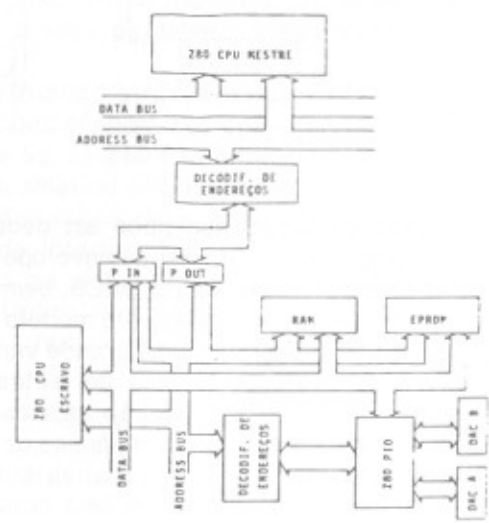


Fig. 2: Diagrama Slave

SOFTWARE

São basicamente dois os níveis em que podemos classificar, quanto a natureza e ordem das atividades desenvolvidas pela CPU, a programação desenvolvida para funcionamento do sistema (Fig. 3).

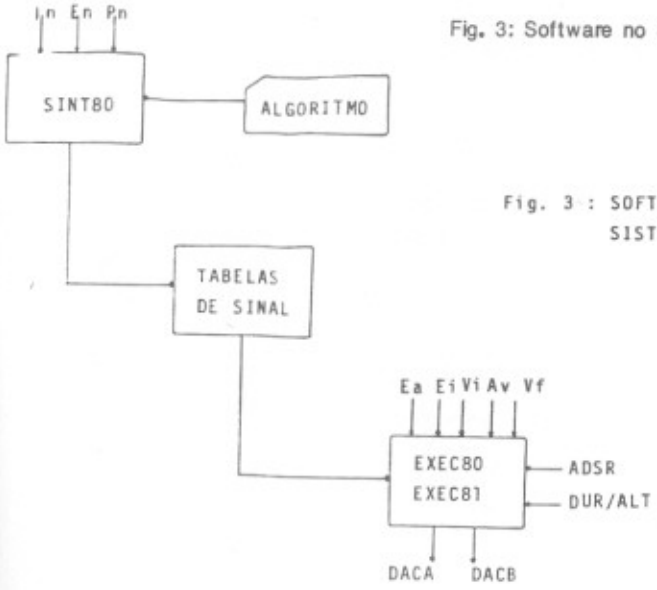


Fig. 3: Software no Sistema

Fig. 3 : SOFTWARE NO SISTEMA

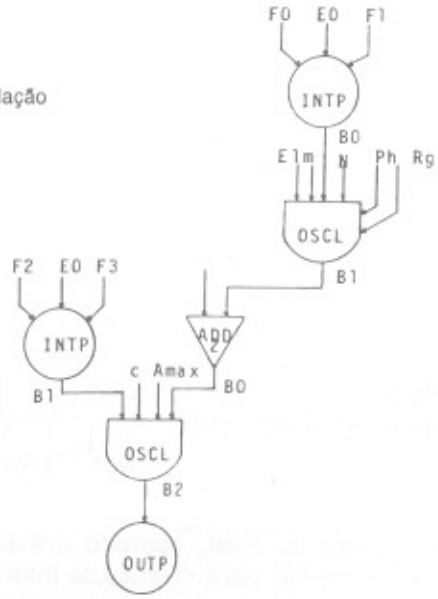
Num primeiro nível, operado previamente à execução de notas, situam-se as rotinas para criação de instrumentos expressos por tabelas de formas de onda ou envelopes lançados na memória.

O software citado (SINT-80) constitui-se de rotinas para geração ou transformação de sinais e, em bloco, vem a formar uma linguagem para criação de timbres ou processamento de sinais digitais voltado à música eletrônica. Foi projetado segundo o princípio modular de unidades geradoras com base no modelo para computador grande porte em que nos inspiramos, o MUSIC V (Mathews, 1981). O programa SINT-80 necessita como dados de entrada funções, envelopes, e parâmetros como Rg ("range") e Ph ("phase"). Através da execução de um algoritmo, fornece como dados de saída as tabelas de sinal ou formas de onda armazenadas na "low-memory" do MSX vindo a constituir os instrumentos para execução.

No programa foram incorporadas unidades como OSCL (oscilador),

INTP (interpolador de espectros), ADD2 (somador de sinais com duas entradas), ADD4 (somador de quatro entradas), FLT1 (rotinas implementando filtro de estado variável), OUTP (transferência de páginas de 256 bytes para a região de armazenamento) que podem ser combinadas diferentemente num algoritmo (executado por uma rotina principal), de forma a implementar variadas técnicas ou combinação de técnicas de síntese. Na Fig. 4 pode ser observada a implementação da modulação de freqüência combinada de forma a realizar interpolação de espectros na evolução da portadora ("carrier") e modulante (Chowning, 1973).

Fig. 4: Algoritmo FM c/Interpolação



Um conjunto de técnicas que abrange a Síntese Aditiva, Síntese Subtrativa (filtros), Modulação de Freqüência, Interpolação, geração de ruído e suas combinações em algorítmo são portanto possíveis no sistema. Outras rotinas úteis na implementação de reconhecidas técnicas de síntese como Wave-shaping, ou análise de timbres via FFT, encontram-se em desenvolvimento. Recursos como a superposição de instrumentos criados e armazenados em disco, foram implementados de forma a ser possível a criação de um novo timbre a partir de dois ou mais originais algoritmos.

O Assembly Z-80 foi escolhido na implantação da programação

por razões de velocidade. Para viabilizar a programação de unidades geradoras mais complexas, mantendo a rapidez de cálculo que vimos obtendo, planejamos a utilização da linguagem C na criação de novas rotinas para o sistema.

Para a geração de funções resultantes da soma de senóides (soma Fourrier), foi implementado um programa onde selecionando-se com um "mouse" os harmônicos desejados, é impressa na tela a função resultante que fica disponível para utilização como entrada em algoritmos de síntese. Através do "mouse" podemos ainda criar manualmente as formas mais variadas de envelope ou funções de sinal que desejemos.

Num segundo nível estão as rotinas dedicadas à geração do som e que operam em tempo-real. Existem atualmente duas versões para os programas de execução no sistema (EXEC-80, 81), onde a primeira dedica-se à execução via teclado do MSX.

O timbre, no momento da execução, é determinado pelo conjunto de tabelas de sinal da "low-memory", tipos de envelope que devem ser acessados de outra área da memória e pelo modo de acesso que é realizado durante a execução. Ao todo são cinco tipos de tabelas acessadas durante a execução: envelope de amplitude (EA), envelope para índice de modulação (EI), tabelas para vibrato (VI) e aceleração no vibrato (Av) e tabela para alteração na freqüência (VF) possibilitando vibratos e glissandos (Fig. 3).

Como foi citado anteriormente, estas rotinas encontram-se implementadas em linguagem Assembly Z-80 e adaptadas ao padrão MSX. Um programa para edição de partitura - texto encontra-se desenvolvido em Basic disco MSX.

REFERÊNCIAS BIBLIOGRÁFICAS

ALLES, Harold G. "Music Synthesis Using Real Time Digital Techniques". Proceedings of the IEEE. 68(4):436-449, April 1980.

CHOWNING, John. "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation". Journal of the Audio Engeneering Society. 21(7):526-534, Sept. 1973.

LOY, Gareth. and Abbot, C., "Programming Languages for Computer Music Synthesis, Performance, and Composition". ACM Computing Surveys. 17(2), june 1985.

MATHEWS, Max V., The Technology of Computer Music. 5th Printing. Massachusetts. MIT Press, 1981, 188pp.

MOORE, F.R. "Table Lookup Noise for Sinusoidal Digital Oscillators". Foundations of Computer Music. 1985, Massachusetts, MIT Press.