

# Interface Gráfica para o Decoder do SoC-SBTVD

Matheus Schuh<sup>1</sup>, Marcelo Negreiros<sup>2</sup>, Altamiro Susin<sup>3</sup>

<sup>1</sup> Autor, Engenharia da Computação, UFRGS

<sup>2</sup> Co-orientador, Engenharia Elétrica, UFRGS

<sup>3</sup> Orientador, Engenharia Elétrica, UFRGS



**XXV SIC**  
Salão Iniciação Científica



ENG - Engenharias

## 1. Introdução

O projeto SoC-SBTVD em desenvolvimento no LaPSI objetiva a criação de um terminal de acesso (set-top-box) para o Sistema Brasileiro de Televisão Digital. Em parceria com outras universidades brasileiras, visa-se construir um SoC (System on a Chip) que contere todos os elementos necessários ao set-top-box, entre eles os decodificadores de áudio e vídeo, circuitos de interface, processador e memória.

Uma parte importante do terminal de acesso é a interface com o usuário, visto que é uma das funcionalidades com a qual ele terá mais contato durante a utilização deste sistema. Ela engloba tanto o controle de periféricos, como o controle remoto, quanto o de componentes internos, como o sintonizador dos canais, por exemplo. Além disso, todas as informações gráficas de configurações, canais e volume são parte desta interface.

## 2. Ambiente de Desenvolvimento

A implementação da parte de software da interface foi no Leon3 (Figura 1), um processador de 32 bits, baseado na arquitetura SPARC V8. O Leon é de código aberto, sintetizável, descrito em VHDL (VHSIC Hardware Description Language), capaz de ser configurado de acordo com a necessidade, acrescentando ou retirando componentes fornecidos por uma biblioteca do desenvolvedor [1]. O protótipo do set-top-box utiliza um sistema de desenvolvimento baseado em FPGA (Virtex 5 da Xilinx [2]) onde estão sendo integrados este processador e os demais componentes do terminal de acesso.

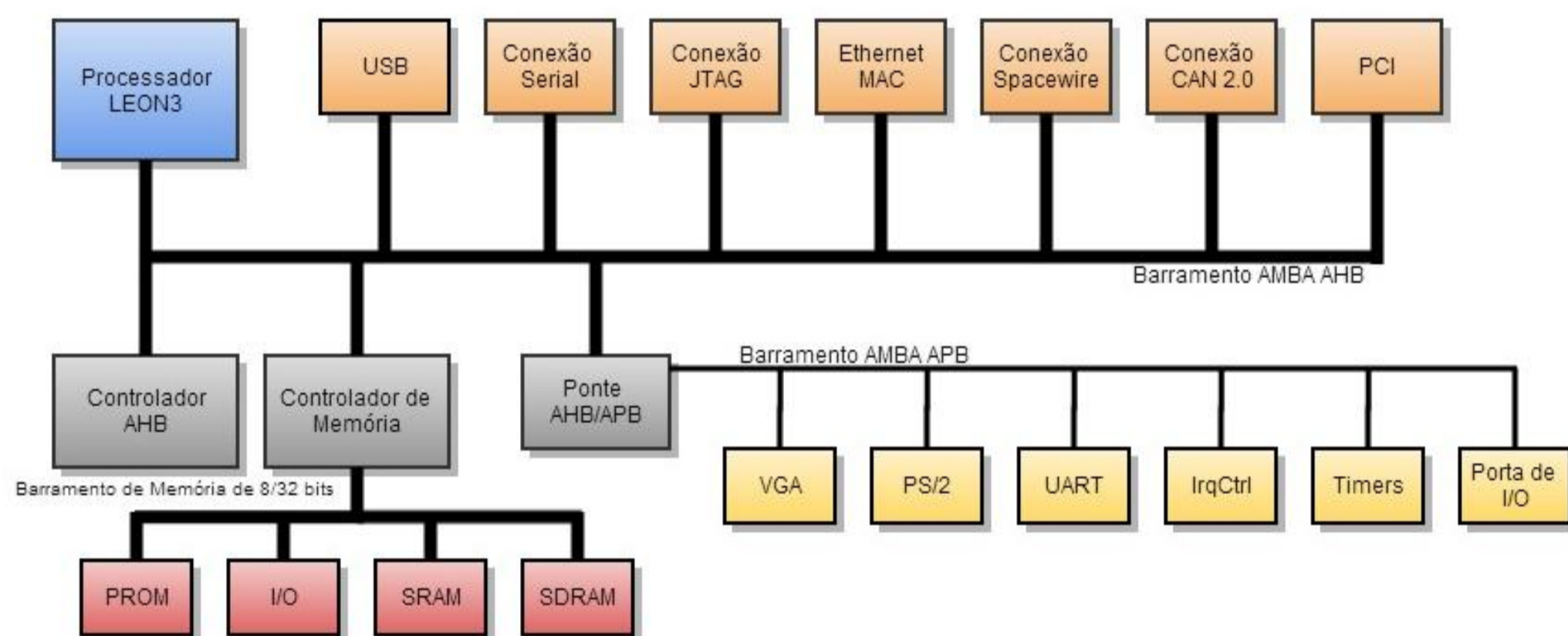


Figura 1. Esquemático do Processador Leon3 e de suas possibilidades de expansão.

## 3. Bibliotecas e Estruturas desenvolvidas

Para implementar o software da interface com o usuário foram desenvolvidas bibliotecas de programação (pode-se acompanhar seus nomes e hierarquias na Figura 2): uma para controle de funções de entrada e saída, como o apertar de um botão ou o acendimento de um LED (*Biblioteca de I/O*), uma de primitivas gráficas que trata de funções primordiais como a mudança de cor de um pixel ou o desenho de uma linha (*Biblioteca Gráfica*) e, por fim, uma voltada à construção de menus para a sinalização de aumento de volume ou troca de canal, por exemplo (*Biblioteca de Menus*).

O programa final (*Aplicação*) utiliza uma biblioteca adicional (*Controle dos Menus*) que realiza a ligação entre o dispositivo de I/O e a parte gráfica. Ela delimita em um nível mais alto as ações de acordo com a interação do usuário permitindo futuras alterações de periféricos.

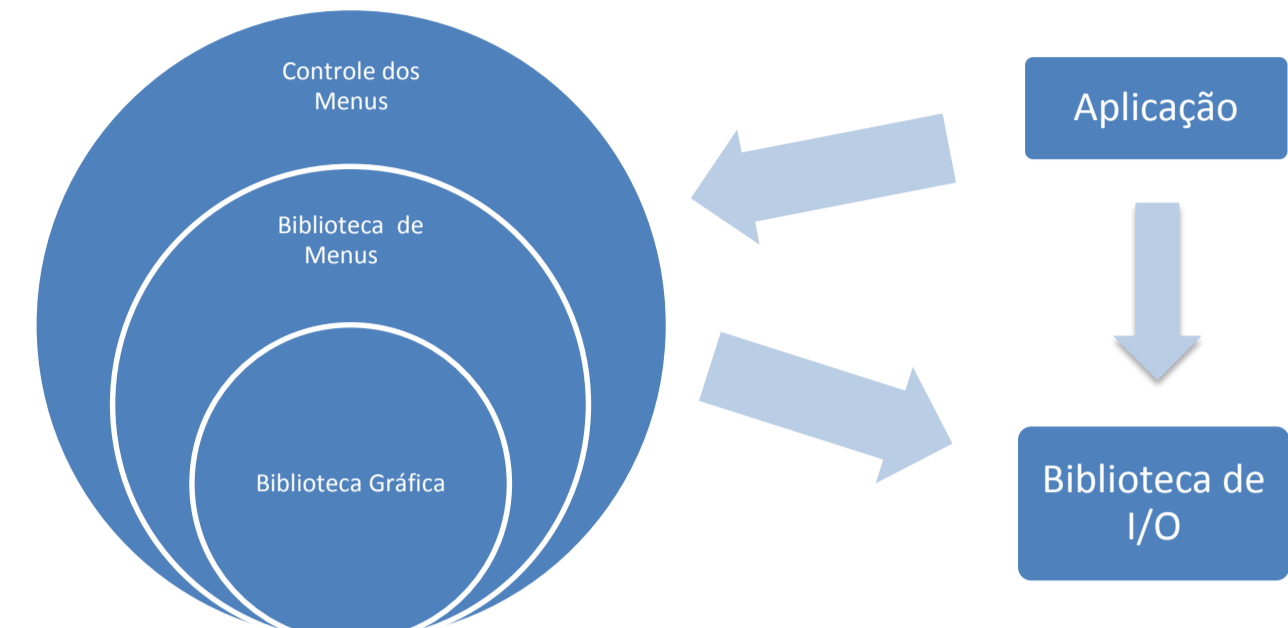


Figura 2. Relação entre as bibliotecas desenvolvidas e o programa principal.

## 4. Resultados

O software foi escrito em C e foi usado um compilador específico (cross-compiler) para a arquitetura do Leon. O arquivo executável é carregado diretamente, sem um sistema operacional, com o auxílio da ferramenta GRMON [3] fornecida pela Gaisler. Incorporado ao projeto do Leon está um decodificador que gera a imagem vista ao fundo da tela nas figuras abaixo. A camada gráfica é sobreposta, portanto, às imagens decodificadas e totalmente independente destas. Por fim vale destacar que a aplicação, mesmo utilizando-se de bibliotecas gráficas próprias e ainda não totalmente otimizadas obteve, para todas as simulações de menus, desempenho praticamente idêntico ao de menus incorporados a aparelhos comerciais atuais.

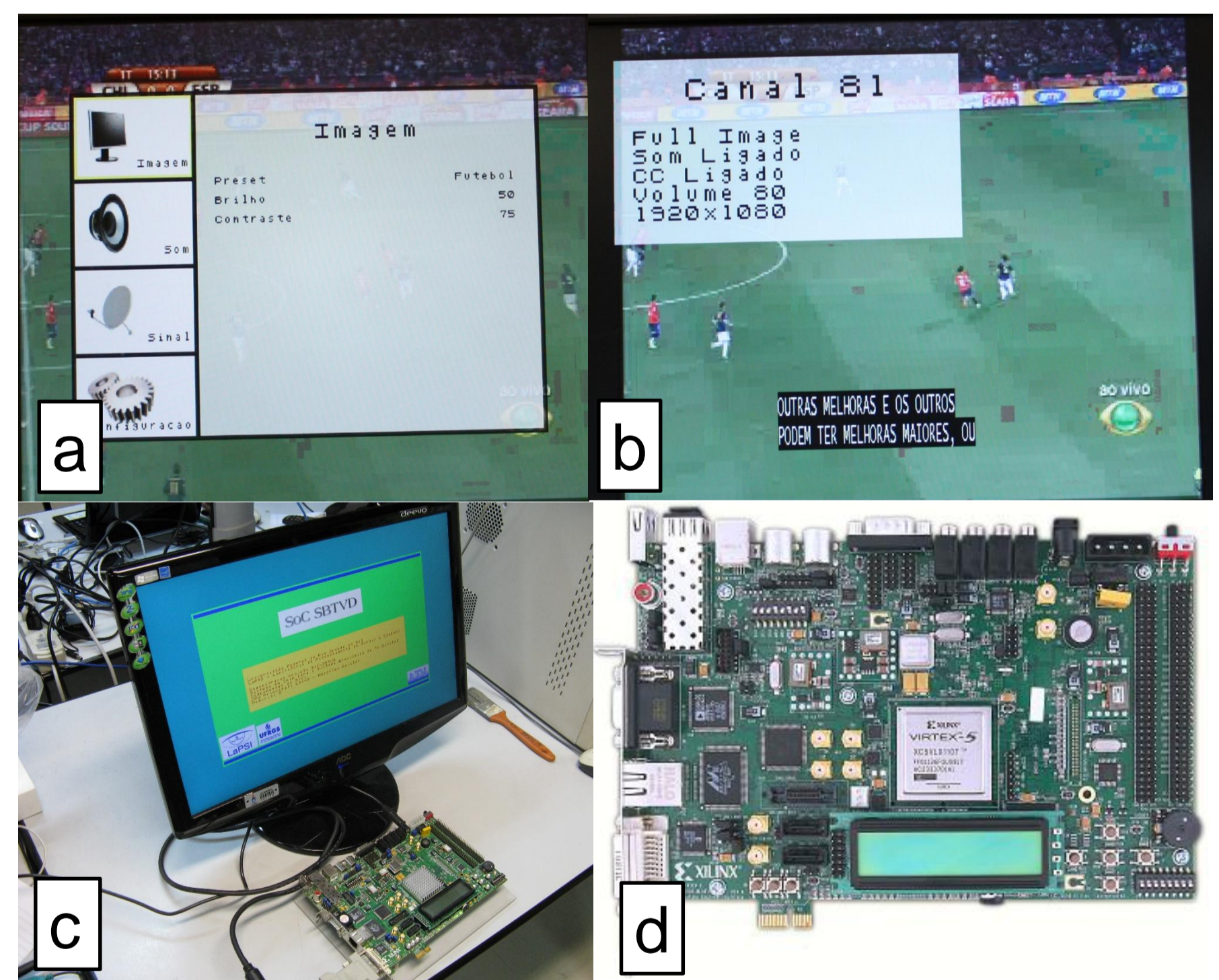


Figura 3 (a), (b), (c) e (d). Demonstração das funcionalidades do sistema. Na figura 3(a) vemos o menu principal da aplicação e em 3(b) a exibição de informações referentes ao canal e à situação do sistema. Abaixo na figura 3(c), uma figura do sistema rodando em uma monitor e em 3(d) uma imagem da placa Virtex-5 da Xilinx usada para sintetizar o Leon juntamente com o decodificador e o software gráfico.

## 5. Referências

[1] GRLIB IP core User's Manual, <http://gaisler.com/products/grlib/grip.pdf>

[2] Virtex5, Overview [http://www.xilinx.com/support/documentation/data\\_sheets/ds100.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf)

[3] GRMON2 Manual, <http://www.gaisler.com/doc/grmon2.pdf>.



MODALIDADE  
DE BOLSA

Iniciação Científica